

# UC San Diego

## UC San Diego Electronic Theses and Dissertations

**Title**

A 16-Channel Electrical Impedance Tomography System Using the Red Pitaya

**Permalink**

<https://escholarship.org/uc/item/7969r8kb>

**Author**

Schluchter, Andrew

**Publication Date**

2020

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA SAN DIEGO

A 16-Channel Electrical Impedance Tomography System Using the Red Pitaya

A Thesis submitted in partial satisfaction of the requirements for the degree Master of Science

in

Bioengineering

by

Andrew J. Schluchter

Committee in charge:

Professor Todd Coleman, Chair  
Professor Pedro Cabrales  
Professor Elliot McVeigh

2020



The Thesis of Andrew J. Schluchter is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

---

---

---

Chair

University of California San Diego

2020

## TABLE OF CONTENTS

Signature Page.....	iii
Table of Contents.....	iv
List of Abbreviations.....	vi
List of Figures.....	vii
List of Tables .....	ix
Acknowledgements .....	x
Vita.....	xi
Abstract of the Thesis .....	xii
1. Introduction.....	1
1.1. Medical Imaging.....	1
1.1. Basics of Medical Imaging and Reconstruction.....	1
1.1.2. Electrical Impedance Tomography as a Modality.....	1
1.2.3. Applications.....	2
1.2. 16-Channel Electrical Impedance Tomography Using the Red Pitaya.....	3
2. Mathematics and Theory.....	6
2.1. Electrical Impedance Tomography Basic Theory.....	6
2.2. Image Reconstruction Theory.....	7
3. Red Pitaya EIT System.....	8
3.1. System Overview.....	8
3.2. Red Pitaya.....	11
3.3. VCCS.....	16
3.4. Multiplexers.....	20
3.5. Tanks and Cables.....	22

3.5.1. Point-Electrode Tank.....	22
3.5.2. Plate-Electrode Tank.....	23
3.5.3. Cables.....	24
3.6. Amplifiers and Filters.....	25
4. Software.....	35
4.1. Software Overview.....	35
4.2. Data Acquisition.....	36
4.3. Image Reconstruction.....	37
5. Methods.....	39
6. Results.....	45
7. Evaluation.....	47
7.1. System Advantages .....	47
7.2. Limitations.....	49
8. Future Work.....	50
8.1. Electrodes.....	51
8.2. Digital Multiplexing.....	51
8.3. Real-Time Microcontroller Processing.....	51
9. Conclusions.....	51
Appendix.....	53
References.....	70

## LIST OF ABBREVIATIONS

**MRI** Magnetic Resonance Imaging

**CT** Computed Tomography

**EIT** Electrical Impedance Tomography

**PET** Positron Emission Tomography

**SPECT** Single-Photon Emission Computed Tomography

**RP** Red Pitaya

**RF** Radio Frequency

**RMS** Root mean square

**SCPI** Standard Commands for Programmable Instrumentation

**VCCS** Voltage-Controlled Current Source

**IA** Instrumentation Amplifier

**CMMR** Common Mode Rejection Ratio

**MUX** Multiplexer

**SNR** Signal-to-Noise Ratio

**EIDORS** Electrical Impedance Tomography and Diffuse Optical Tomography

**MATLAB** Matrix Laboratory

**FEM** Finite Element Model

**FE** Finite Element

**ROI** Region of Interest

## LIST OF FIGURES

Figure 1.1. Example EIT Images of Goe-MF II EIT system.....	2
Figure 1.2. Basic Process from Physical Signal Acquisition to Reconstruction.....	5
Figure 3.1. System Power Supply.....	9
Figure 3.2. Photo of All Data Acquisition Components.....	10
Figure 3.3. Simplified Schematic of All Data Acquisition Components of EIT System.....	11
Figure 3.4. Red Pitaya with All Connections.....	12
Figure 3.5. RP Mini-Tank Testing Setup.....	13
Figure 3.6 RP Signal Testing.....	15
Figure 3.7. VCCS Design, Simulation, and Simulation Response.....	17
Figure 3.8. Photo of the VCCS Circuit Physical Construction.....	19
Figure 3.9. VCCS Circuit Output Current Response with Variable Frequency and Output Loads.....	20
Figure 3.10. Photo of Multiplexer Array.....	21
Figure 3.11. Point-Electrode Tank, with Object.....	23
Figure 3.12. Plate-Electrode Tank.....	24
Figure 3.13. Custom Cables for EIT Tanks.....	25
Figure 3.14. Instrumentation Amplifier with Attached DC-Removing Integrator Circuit.....	27
Figure 3.15. Schematic Template Used for Output Amplifier.....	28
Figure 3.16. Schematics of Amplifier, with Testing.....	29
Figure 3.17. Photo of All Amplifiers and Filters, Labelled.....	30
Figure 3.18. Results of Output Amplifier Across Various Frequencies and DC Offsets.....	31
Figure 3.19. Photo of All Signal Circuits Together.....	32
Figure 3.20. Full System Schematic.....	33
Figure 3.21. Photo of Plate-Electrode Tank, with 2 Objects.....	34



Figure 3.22. Example Data to Reconstructed Image.....	34
Figure 4.1. Data Acquisition Flowchart.....	36
Figure 4.2. Various K-Values used in Reconstructing Images with the Point-Electrode Tank.....	39
Figure 5.1. Testing the point-electrode tank object reconstruction accuracy in 8 example positions.....	41
Figure 5.2. Point-electrode tank object reconstruction in 8 example positions.....	42
Figure 5.3. Point-electrode tank object reconstruction after thresholding, in 8 example positions.....	44
Figure 5.4. Plate-electrode tank thresholding results.....	46
Figure 5.5. TVR Filtering .....	47

LIST OF TABLES

Table 1. MUX Addressing Table.....22

Table 2. SNR Point-Electrode Tank Table.....45

## ACKNOWLEDGEMENTS

I would like to acknowledge my research partner Victor Bourgin, particularly for some of the coding, and his work in testing and generating data.

I would also like to thank my advisor Dr. Todd Coleman for his financial support and the process of working with his lab, and my PI Dr. Elliot McVeigh for his support of my busy schedule.

## VITA AND PUBLICATIONS

2007-2013 U.S. Navy, Nuclear Electronics Technician, PO2

2015-2016 STEM Tutor, Old Dominion University

2016 Bachelor of Science in Electrical and Computer Engineering, Old Dominion University

2016 Bachelor of Science in Biology, Old Dominion University

2016-Present Lab Manager at Cardiovascular Imaging Lab (CViL), University of California San Diego

2020 Master of Science in Bioengineering, University of California San Diego

## PUBLICATIONS

Schluchter, A., Jan, C., Lowe, K., Vigneault, D. M., Contijoch, F., & McVeigh, E. (2019). Vascular Landmark-Based Method for Highly Reproducible Measurement of Left Atrial Appendage Volume in Computed Tomography. *Circulation: Cardiovascular Imaging*, 12(12), e009075-.

Manohar, A., Colvert, G., Schluchter, A., Contijoch, F., & McVeigh, E. R. (2019, March). LV systolic point-cloud model to quantify accuracy of CT derived regional strain. In *Medical Imaging 2019: Image-Guided Procedures, Robotic Interventions, and Modeling* (Vol. 10951, p. 109510E). International Society for Optics and Photonics.

Chen, Z., Contijoch, F., Schluchter, A., Grady, L., Schaap, M., Stayman, W., Pack, J. and McVeigh, E. (2019). Precise measurement of coronary stenosis diameter with CCTA using CT number calibration. *Medical physics*, 46(12), 5514-5527.

Colvert, G. M., Manohar, A., Colvert, B., Schluchter, A., Contijoch, F., & McVeigh, E. (2019, March). Novel measurement of LV twist using 4DCT: quantifying accuracy as a function of image noise. In *Medical Imaging 2019: Biomedical Applications in Molecular, Structural, and Functional Imaging* (Vol. 10953, p. 109531K). International Society for Optics and Photonics.

## ABSTRACT OF THE THESIS

### A 16-Channel Electrical Impedance Tomography System Using the Red Pitaya

by

Andrew J. Schluchter

Master of Science in Bioengineering

University of California San Diego, 2020

Professor Todd Coleman, Chair

Electrical Impedance Tomography (EIT) seeks to provide a new modality by which to image portions of the body and tissues which provide differences in conductivity, depending on their state. Unlike bulky, expensive, and hard-to-access traditional medical imaging equipment, such as those used in magnetic resonance imaging and computed tomography, EIT is potentially capable of being contained to small, portable, inexpensive hardware. Here, a Red Pitaya (RP) device is used to provide and process signals that can be generated and multiplexed into 16 channels, all while data is gathered from a set of electrodes embedded into a tank containing electrically conductive simulated tissue. A voltage-controlled current source provides a known value of current to be introduced with each new set of measured signals, which are then amplified and filtered to preprocess the signals. The data is then gathered together by the RP and communicated to a MATLAB program running on a nearby PC through a Standard Commands for Programmable Instrumentation interface, where the data is reconstructed using

regularization with EIDORS, a MATLAB toolkit. Precision and accuracy were measured by evaluating the signal-to-noise-ratio (SNR) and a 2D Pearson correlation coefficient to the ideal images of the tank. The first tank design used electrodes of a small, pointed design, while the second used 1 cm<sup>2</sup> copper plates for electrodes. The first had a mean system SNR of 31.616 dB (11.347) before filtering and 34.176dB (11.9803) after, and a correlation coefficient of  $r = 0.702$  (0.055), while the second tank showed negative results.

## **1. Introduction**

### **1.1. Medical Imaging**

#### **1.1. Basics of Medical Imaging and Reconstruction**

Medical imaging and reconstruction is a crucial part of clinical research and the identification and treatment of many diseases and injuries. Currently, the most prevalent forms of imaging in practice are ultrasound, Magnetic Resonance Imaging (MRI), X-ray (including Computed Tomography, or CT), and nuclear medicine (such as SPECT and PET). In all modalities, energy is deposited into a volume of tissue, and the attenuated energy is captured, recorded, and processed into a usable form [1]. From here, reconstruction can be performed.

All deposited energies come at some form of cost. In techniques involving radiation, some amount of radiation dose is given to the patient, which must be monitored and minimized to prevent over-exposure, which can lead to disease, particularly cancer [2]. In MRI, the acquisition of images involves a long exposure time as well as very expensive machinery, limiting the practical use to patients and researchers [3]. Even techniques which do not expose patients to radioactive energies or length times for acquisition still deposit energy. In ultrasound, excessive heat can be generated from the oscillation of sound waves in the tissue, which can even be high enough to damage tissue [4].

#### **1.1.2. Electrical Impedance Tomography as a Modality**

Other properties inherent to materials and tissues can be found in their abilities to respond to electrical stimulation, such as their electrical conductivities. In tissue, conductivity changes can indicate abnormalities, such as inflammation, contusion, air pockets, and tumor growth [5]. This change in conductivity is the basis for electrical impedance tomography (EIT). One commercial device is the Goe-MF II EIT system [6]. This is a professional, EIT system which can be used clinically to monitor lung function and esophageal disorders.

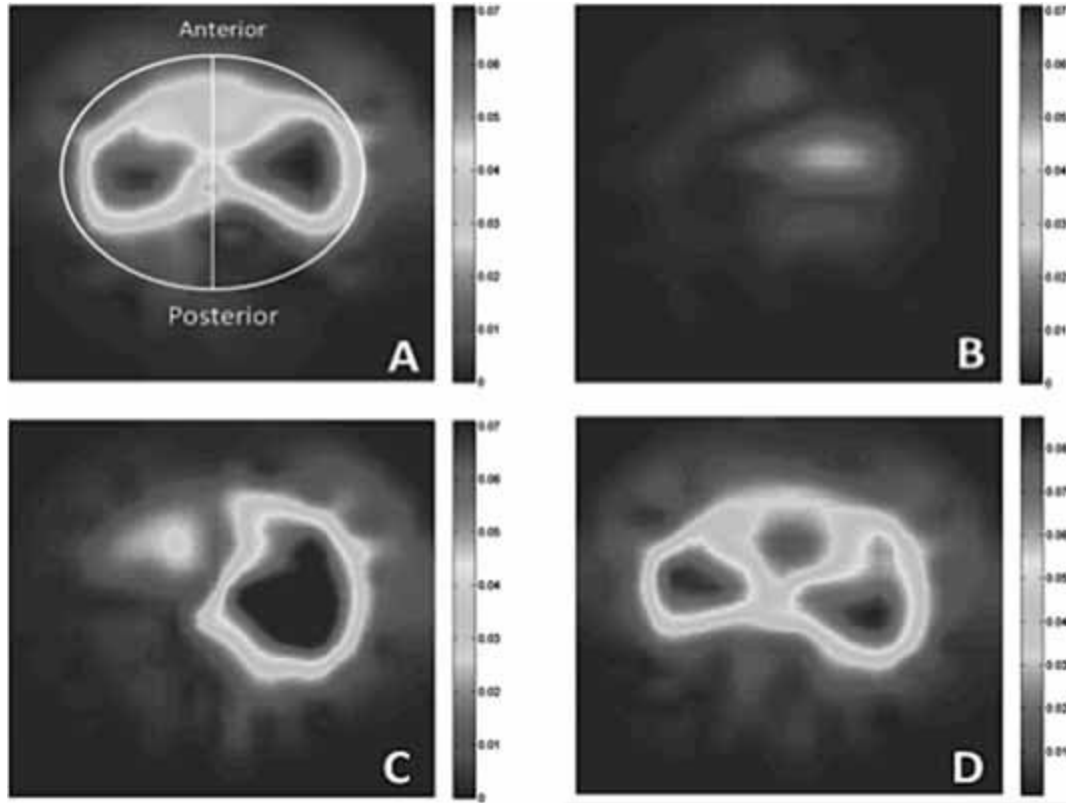


Figure 1.1. Example EIT Images of Goe-MF II EIT system. Lung function is shown using a band of electrodes around the patient's chest [6].

Unlike the previously mentioned modalities, EIT data can be collected using voltages low enough to not create excessive electrical currents, and currents low enough to not result in tissue damage [7]. Radiation exposure is removed from the acquisition process, as are extreme magnetic field exposure, further preventing complications during acquisition. Lastly, reconstruction can be performed on consumer-grade computer equipment reasonably quickly.

### 1.2.3. Applications

Each medical imaging modality is also limited to perceiving only a subset of medical conditions and tissue types. For instance, while CT may excel at visualizing large masses and functional abnormalities, and MRI may be better at visualizing smaller masses and certain types of functionality, cancerous and inflamed tissue may still be difficult, if not impossible to observe [1]. In ultrasound, even greater limitations are present, such as high SNR, and limited resolution.



Unlike all other modalities, EIT is able to utilize tissue impedance to visualize structures, resulting in entirely different types of visualizations. Some applications may include gastrointestinal imaging, esophageal disease detection, and new forms of disease characterization [8],[11]. Because commercial EIT devices are still fairly new (relative to other commercially available medical imaging equipment), many new applications may be found as different diseases and tissue types are imaged using this modality.

Reconstruction of images from the data collected can pose a particularly difficult mathematical problem, as the problem itself is not a well-posed problem. That is, because the electrical path of conduction between electrodes is non-linear, and because there are a finite number of boundary conditions, some form of regularization is required in reconstruction, which is expanded on in the section Mathematics and Theory [9].

The commercial systems currently in use, such as the Goe-MF II [10] or the PulmoVista 500 [11], show that there is merit and practical use in improving medical imaging via EIT, with a focus on maximizing the benefits of EIT while minimizing or resolving the limitations. In the case of EIT, electrical stimulation and current detection can theoretically be performed on low-cost, portable equipment, and the reconstruction may be done on most modern computing hardware. Thus, portability and compactness should be highlighted, and focus should remain on obtaining the most noise-free data possible, while optimizing the quality of the reconstructed data.

One portable EIT system has already been developed by Xu [12]. It uses 8 channels, and the data is collected and processed by a Red Pitaya (RP) device. This device is a consumer or hobbyist-grade open-source RF device, and is able to be programmed by the user [13]. The system was created using only 8 channels, however, which limited the feasibility of its use in acquiring clinically-relevant images. In our system, we expand on the work by increasing the number of channels using a multiplexing system, and by comparing electrode types using different tanks.

### **1.2. 16-Channel Electrical Impedance Tomography Using the Red Pitaya**

The RP as an EIT processor relies on a complex setup made from different subsystems. The basic process to go from signal to image is shown in Figure 1.2. The tank is used for injecting current signals and for measuring voltage values (which correspond to conductivities). That data is organized into

a series of data points, with each data point corresponding to measured voltage values. This data is then processed through a PC and reconstructed into usable images.

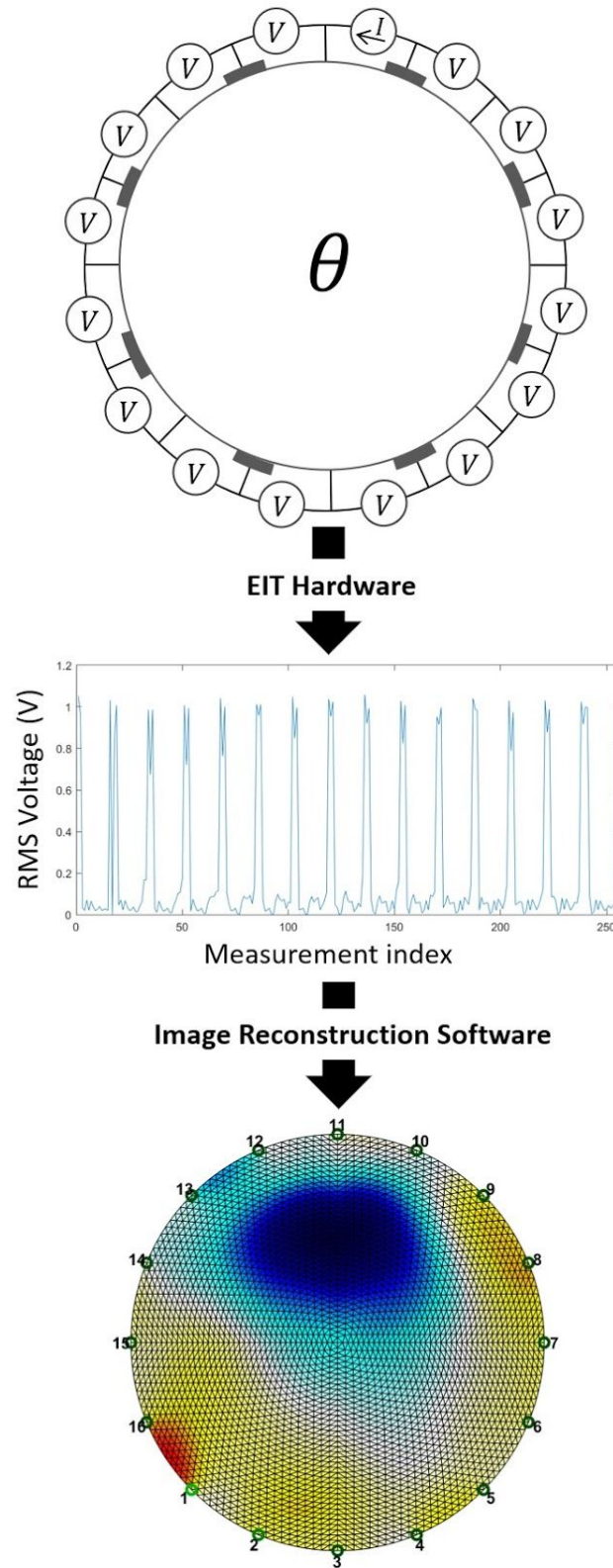


Figure 1.2. Basic Process from Physical Signal Acquisition to Reconstruction

## 2. Mathematics and Theory

### 2.1. Electrical Impedance Tomography Basic Theory

We begin with the simple Ohm's Law equation:

$$E = I R, \quad (1)$$

for an electric field  $E$ , current  $I$ , and a resistance  $R$ . Rather than using simple resistance, we will instead use conductivity, and by expanding voltage into a more general electric potential and assuming no significant currents are generated from the body, we get the equation:

$$J = \sigma E, \quad (2)$$

for a current  $J$  and conductivity  $\sigma$ . Compare this to Maxwell's equation for electromagnetic wave transmission:

$$\nabla \times E = -\frac{\partial B}{\partial t}, \text{ and } \nabla \times B = \mu_0 \epsilon_0 \frac{\partial B}{\partial t}, \quad (3)$$

for a magnetic field  $B$ , permeability of free space  $\mu_0$ , and permittivity of free space  $\epsilon_0$ . Here we can use the following:

$$B = \mu H \quad (4)$$

for permeability  $\mu$  and a magnetizing field  $H$ . We apply the substitutions and recreate in a frequency domain form to get:

$$\nabla \times E = -j\omega\mu H, \text{ and } \nabla \times H = J + j\omega\epsilon E, \quad (5)$$

for the complex number  $j$ , and frequency  $\omega$ . We substitute with (2) to get:

$$\nabla \times H = \sigma E + j\omega\epsilon E = (\sigma + j\omega\mu_0\epsilon)E \quad (6)$$

If we keep our injection signals between 10 and 100 kHz [14], we can assume that the frequency-dependent variables are negligible:

$$\nabla \times H = \sigma E \quad (7)$$

And if we assume that from [14],

$$E = -\nabla u, \quad (8)$$

for  $\nabla u$  as the in-body electrical potential, then:

$$\nabla \times H = -\sigma \nabla u, \quad (9)$$

The divergence of which is:

$$0 = \nabla \cdot (\sigma \nabla u) \quad (10)$$

for a measured set of potentials ( $u$ ), there exist some set of conductivities ( $\sigma$ ). The set of potentials acquired by the EIT system act as the boundary conditions, allowing the conductivities to be solved for. However, this is not a well-posed problem in that many solutions may exist that reflect the boundary conditions. The conductivities are non-linear as well, making the solution(s) more difficult to obtain. Raising the number of channels as high as possible increases the number of boundary conditions, allowing for greater accuracy (though higher processing times).

## 2.2. Image Reconstruction Theory

In this project, we have been able to apply these equations in practice to obtain electric potential measurements from specialized custom tanks, and reconstruct images using software packages designed for EIT image reconstruction. The end result is a picture that approximates the conductivities calculated from the configuration of the tank.

To begin, we assume that a set of voltages (which are our boundary values) are a function of conductivity. For a set of voltages  $V$  within the Region of Interest (ROI),

$$V = f(\sigma), \quad (11)$$

Since we are using differential imaging, we may assume that each conductivity will change from  $f(\sigma)$  to  $f(\sigma + \Delta\sigma)$ , which allows us to use Taylor series expansion to obtain:

$$\begin{aligned} \Delta V &= f(\sigma + \Delta\sigma) - f(\sigma) \\ &= f(\sigma) + f'(\sigma) \cdot \Delta\sigma + \frac{1}{2} f''(\sigma) \cdot \Delta\sigma^2 + \dots + \frac{1}{n!} f^n(\sigma) \cdot \Delta\sigma^n - f(\sigma) \end{aligned} \quad (12)$$

For relatively small changes in the measured conductivities, we can simplify (12) to:

$$\Delta V \approx f'(\sigma_0) \cdot \Delta\sigma \quad (13)$$

By rewriting as a matrix equation, we can say:

$$\Delta \mathbf{V} = \mathbf{J} \cdot \Delta \boldsymbol{\sigma} , \quad (14)$$

for  $\Delta \mathbf{V}$ , a vector of the boundary values (signals observed) and of equal length to the number of electrodes, and for  $\Delta \boldsymbol{\sigma}$ , a vector representing the change in conductivities.  $\mathbf{J}$  is the Jacobian of  $\Delta \mathbf{V} \times \Delta \boldsymbol{\sigma}$ .

Although equation (14) cannot be solved for as is, we can instead recreate it as a Finite Element (FE) model, and if needed, perform filtering through various algorithms such as Total Variation Regularization (TVR) [15] or Tikhonov regularization [16].

To build the model, we discretize our  $\Delta \mathbf{V}$  matrix based on the number of electrodes used by our system (16 channels) using the measured signals (our boundary values). Building the Jacobian  $\mathbf{J}$  is equivalent to the process of solving the Inverse Problem, which is what allows us to go from observed boundary values (the measured potentials) to an approximation of the conductivities within the ROI that could lead to the boundary values [17].

Building the Jacobian requires both linearization (due to the problem's non-linear nature), and an iterative method for approximating the result. In our case, using our previously mentioned assumption of small changes in conductivities as per equation (13), we may use the one-step Gaussian-Newton reconstruction algorithm [18]. This is in lieu of the methods such as the filtered backprojection seen in CT [19].

Additional filtering can also be performed on the end image, to smooth visual artifacts and highlight particular features. This is also especially useful in medical imaging, as although data is often able to be gathered, it must also be clearly interpretable [20].

### 3. EIT Using the Red Pitaya

#### 3.1. System Overview

Our EIT system is based on the Red Pitaya device, made by the company of the same name. The Red Pitaya (RP) is an open-source hardware measurement and control device designed for RF

equipment. In our system, it has been purposed into the central component of our EIT data acquisition system.

Power to the RP is produced by a TP3005D-3M adjustable, dual-voltage DC power supply by TekPower [21]. Power is provided to the amplifying and filtering circuitry at both +5 and -5 volts. The power supply is shown in Figure 3.1.

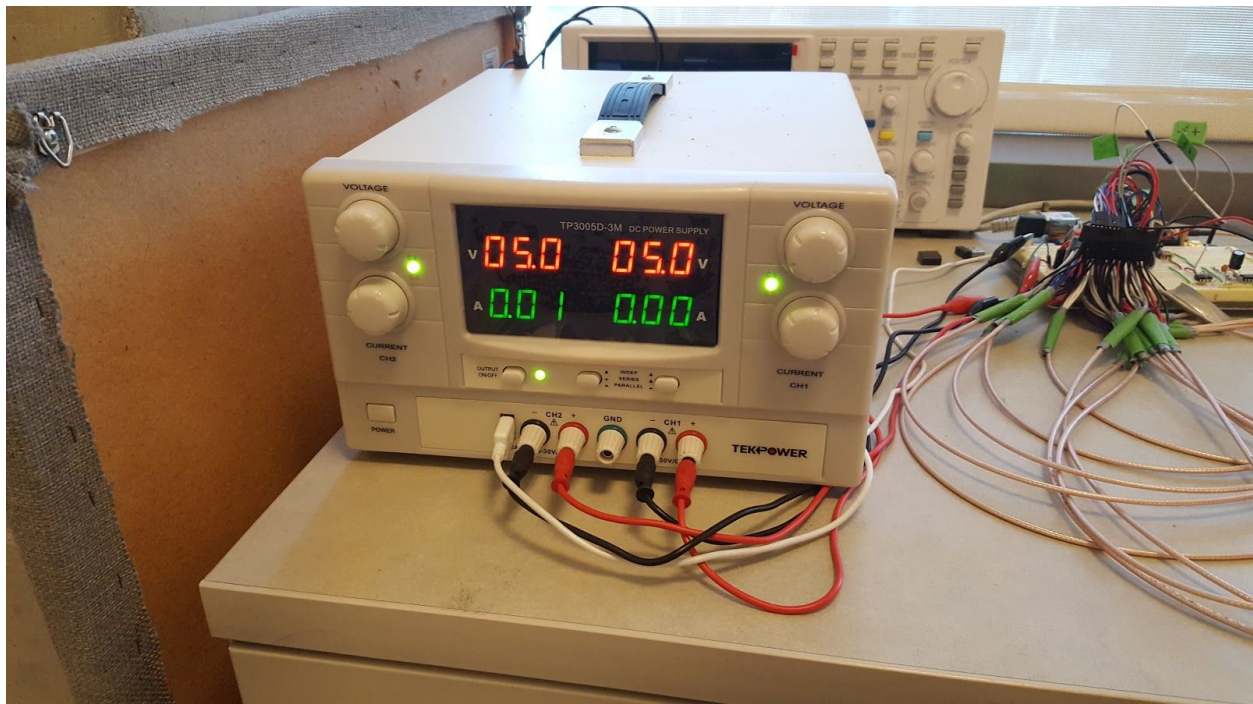


Figure 3.1. System Power Supply

A voltage-controlled current source (VCCS) converts the analog voltage supplied by the RP into a constant current source, which is then able to be distributed by the multiplexers.

The multiplexers (MUX's) are data selection circuits, and allow for each current signal from the VCCS to be connected to each tank electrode, as appropriate. This allows for iterations of injections and measurements to be made without having to physically disconnect and reconnect the electrodes [22].

The basic tank design is a cylinder with an imaging volume of 112mm (diameter) by 25mm height. 16 electrodes built into the sides of the tank allow for the acquisition of 16 channels of data. 2 different designs are used for the tank. The first is a point-electrode tank, made with small, pin-like contacts. The second is a plate-electrode tank, with 1 cm<sup>2</sup> copper plates for the electrode surface. Each

tank is designed with 3D modelling software Blender and printed on a Form 2 3D printer by Formlabs [23], [30].

Lastly, the incoming data is filtered and amplified to condition the signal prior to be received by the RP. The amplification and filtering circuits consist of an instrumentation amplifier, a DC-removing integrator circuit, a band-pass AC filter, and an output amplifier.

Simulation of all circuitry was performed on LTSPICE XVII electronic simulation software, and evaluated for performance [27].

A photo of the full system for data acquisition (not including the PC used for image reconstruction) is shown in Figure 3.2., and the simplified schematic is shown in Figure 3.3.

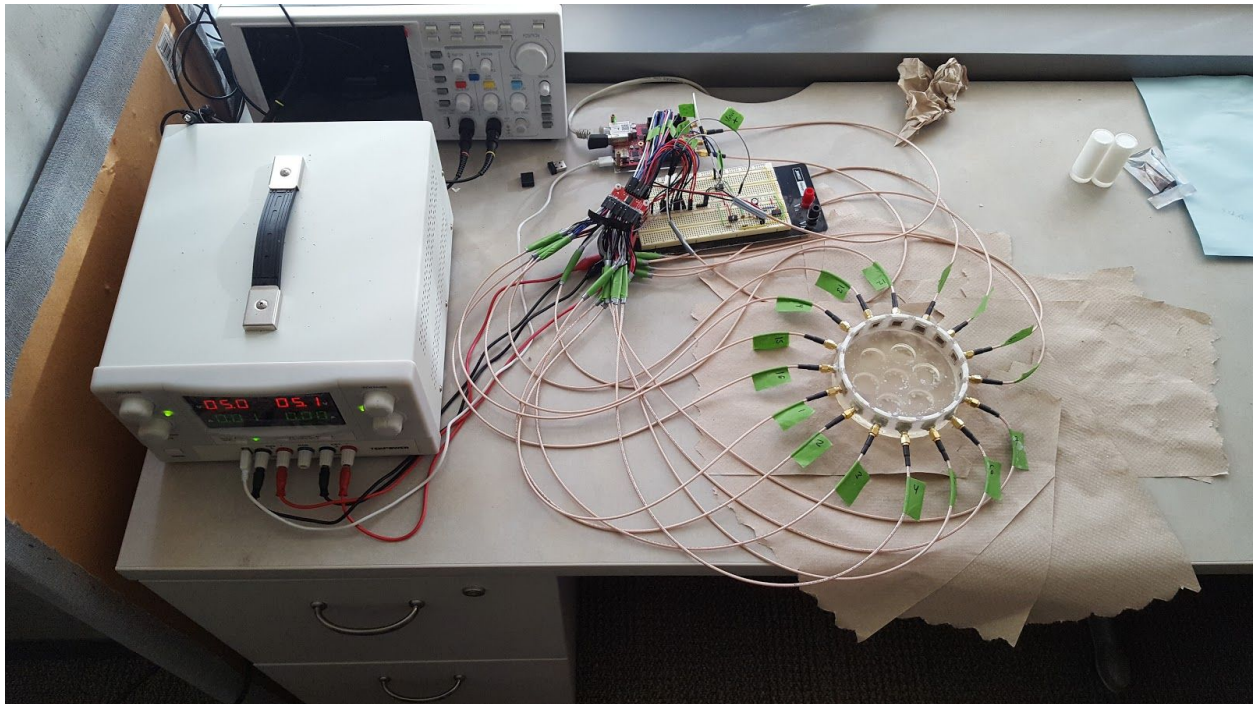


Figure 3.2. Photo of All Data Acquisition Components



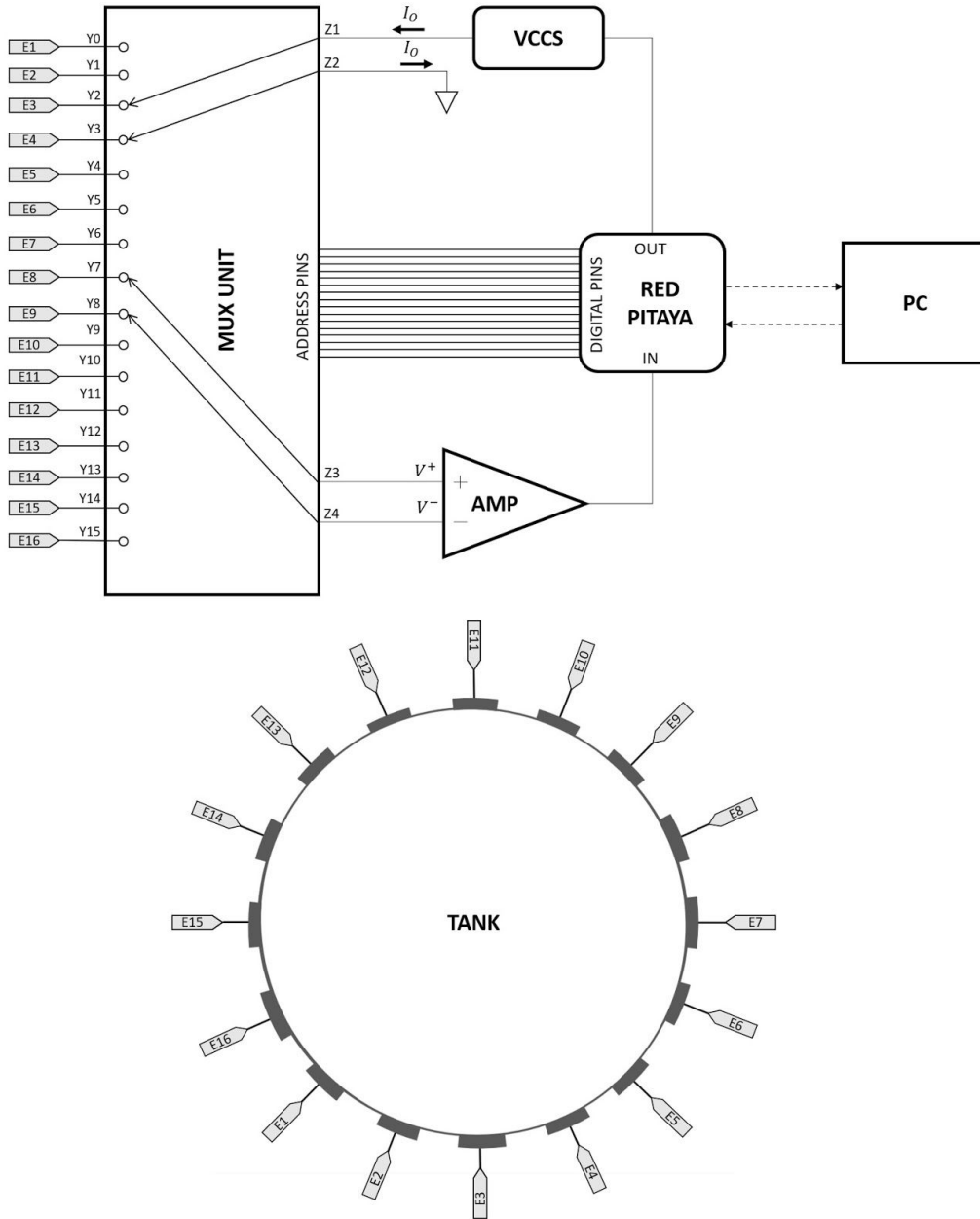


Figure 3.3. Simplified Schematic of All Data Acquisition Components of EIT System

The data is then sent to a nearby PC for data processing and image reconstruction.

### 3.2. Red Pitaya

The RP is a generally open-source device (although the hardware remains proprietary), designed primarily for hardware-based radio frequency (RF) and electronic projects. Onboard analog-to-digital and digital-to-analog converters use 14-bit resolution for high precision. It has two pairs of 50MHz analog input

and output channels, for a total of four analog connections to the board. Maximum output voltage is 1V. Channels can be sampled at up to  $125 \times 10^6$  samples per second [13]. The RP is seen in Figure 3.4. With all connections present.

In addition to the analog channels, the RP has 16 3.3V digital I/O pins, which we use later for addressing circuits that allow us to go from a single analog output channel, to 16 different signals. The RP communicates to external equipment through a variety of different methods. We have chosen to have the RP act as a wireless access point, through which a PC can access as a wi-fi network. From there, commands may be given through MATLAB scripts. Other languages, such as Python, are also supported by the RP.

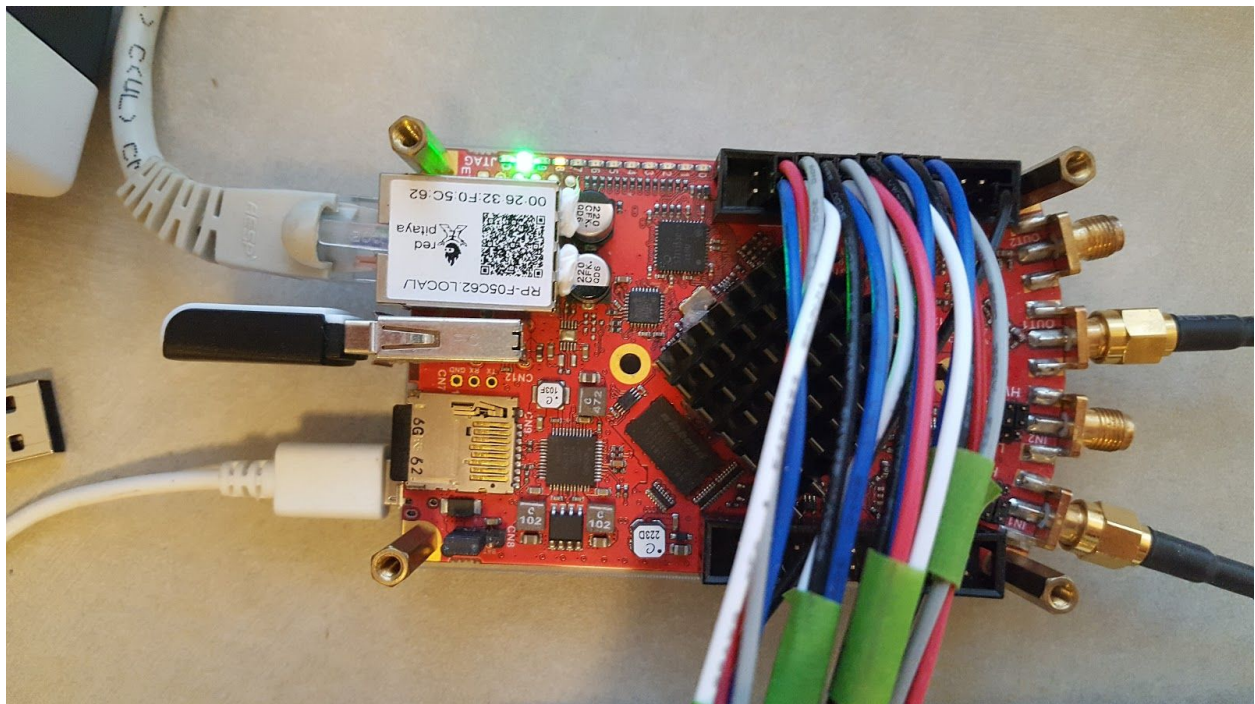


Figure 3.4. Red Pitaya with All Connections. Power is provided through the microUSB cable on the lower left, Wi-Fi is through the USB dongle on the left center, and digital addressing is provided through the connector on the top of the RP as shown. Analog signals are visible on the right.

To test the RP output, we connected the analog output to a small test version of a tank. This miniaturized tank consisted of a 3D printed cylinder with a diameter of 10mm and a height of 12mm, with only two electrode channels. Each electrode was made with a gold-plated SMA RF connector and terminating in the tank volume as a small electrical pin. These are the same used in the “point-electrode

tank” described in section 3.1.4.1. The testing setup is shown in Figure 3.5 using cables provided with the RP.

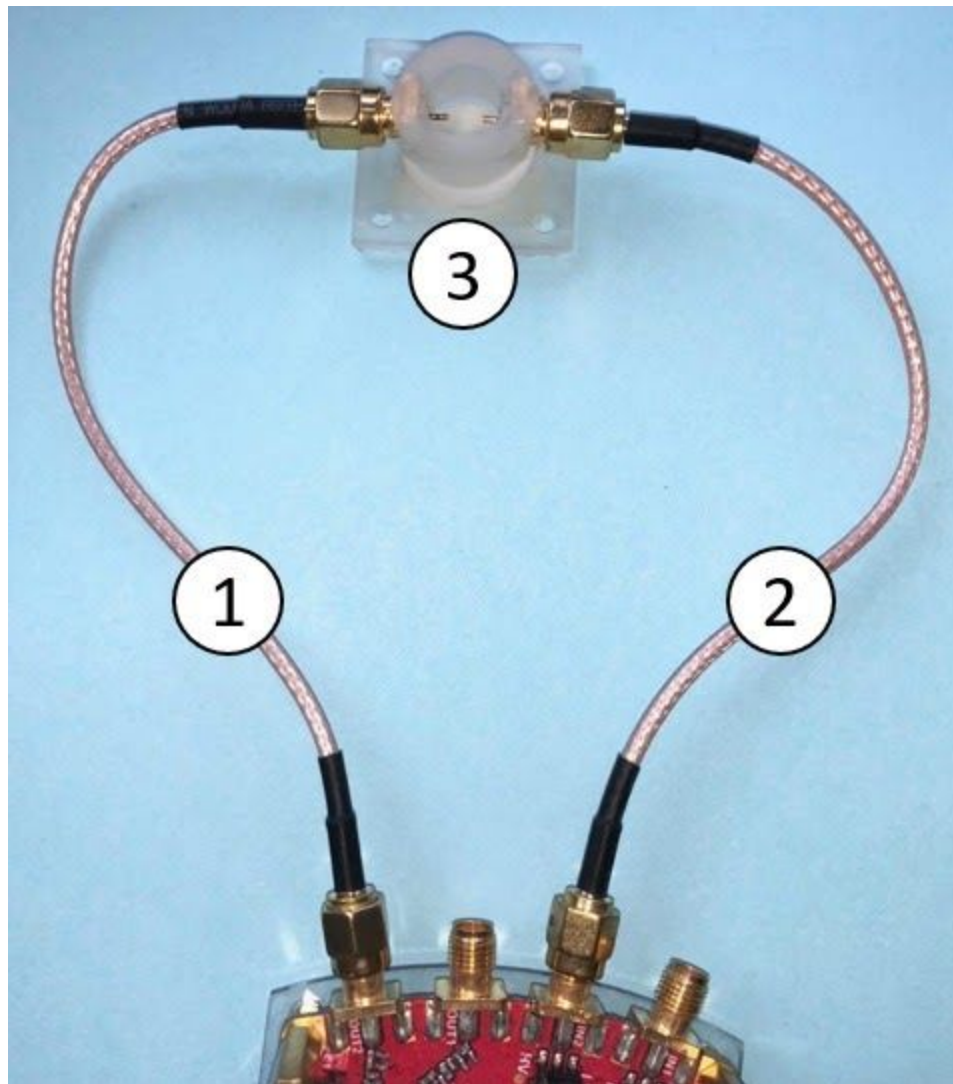
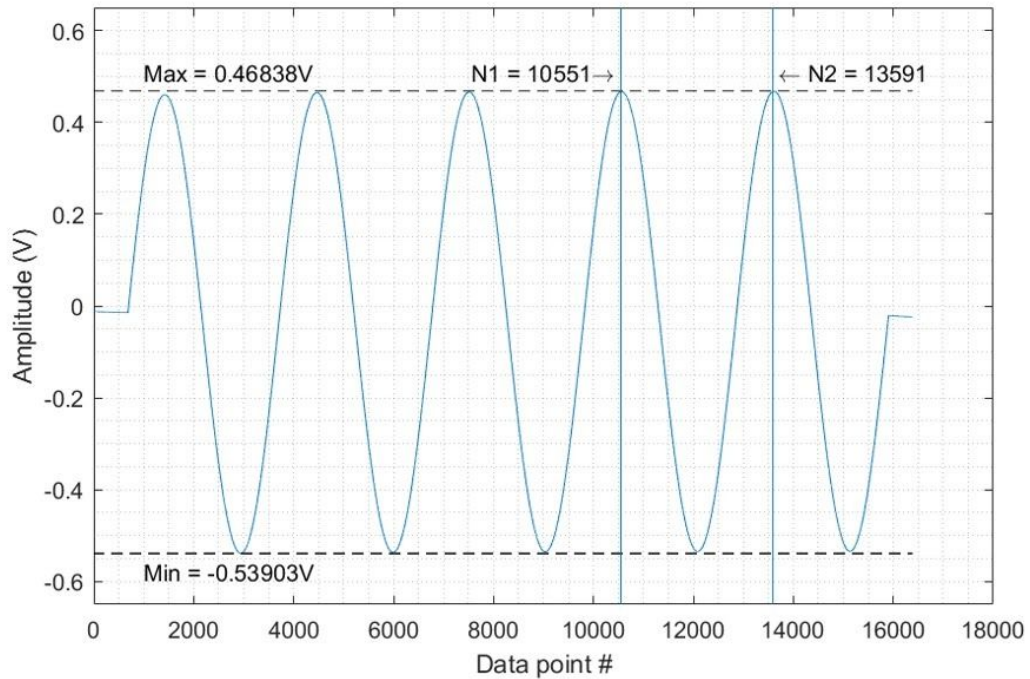


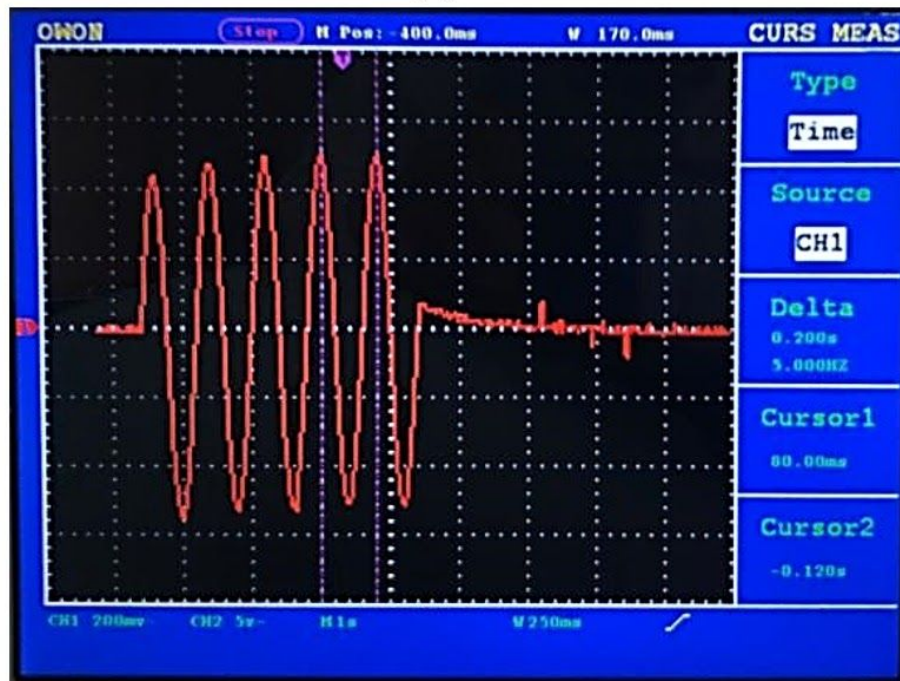
Figure 3.5. RP Mini-Tank Testing Setup

MATLAB code was written to send a 5Hz, 0.5V signal with a length of 5 periods, and is available in the Appendix. The tank was filled with a saturated brine solution (350mg/mL of salt in purified water). Voltage amplitude was verified from the RP to be 0.5037V, and frequency was determined to have a period of 3040 samples. The input was sampled at 15.2588kHz (downsampled from the RP sampling frequency of  $125 \times 10^6$  samples per second at a factor of 8,192). Thus, the frequency seen by the RP was 5.019Hz. This is shown in Figure 3.5.1.

The voltage and frequency response from the mini tank was measured using the RP analog input and using an OWON PDS5022T oscilloscope and verified as expected [24]. Test response is shown in Figure 3.5.2.



(1)



(2)

Figure 3.6. RP signal testing. (1) Test signal as received and analyzed through the RP analog input channel 1. (2) The same test signal as viewed on an oscilloscope.

### 3.3. VCCS

To inject current of known values into the EIT tank, we must convert from the RP's voltage output, to a current output. This is performed by the VCCS as follows. The VCCS is built using an INA105 Precision Unity Gain Differential Amplifier, by Texas Instruments [25]. The schematic is shown in Figure 3.6.1., and the current output equation is:

$$I_O = V_{IN} \left( \frac{1}{25k\Omega} + \frac{1}{R} \right) \quad (15)$$

We must set the output current to a value that is high enough to provide a usable signal, while low enough not to damage tissue. To ensure we meet applicable safety requirements, we refer to IEC 601-1 [26]. We are using frequencies over 1kHz for signal injection, with a bandwidth of 100kHz. Thus, we use the maximum current value limit for frequencies up to 100kHz, which is 100  $\mu$ A(rms).

With the RP maximum output voltage of up to 1V, we wish to set the value of R such that the input voltage  $V_{IN}$  can be in range of the RP output. We set R to 4.6k $\Omega$  for a  $V_{IN}$  of +0.55V to correspond to an  $I_O$  of 141 $\mu$ A, or 100 $\mu$ A (rms) at a frequency of 10kHz. In Methods, we use a  $V_{IN}$  of .1V amplitude at 10kHz for a current output of 25.6 $\mu$ A.



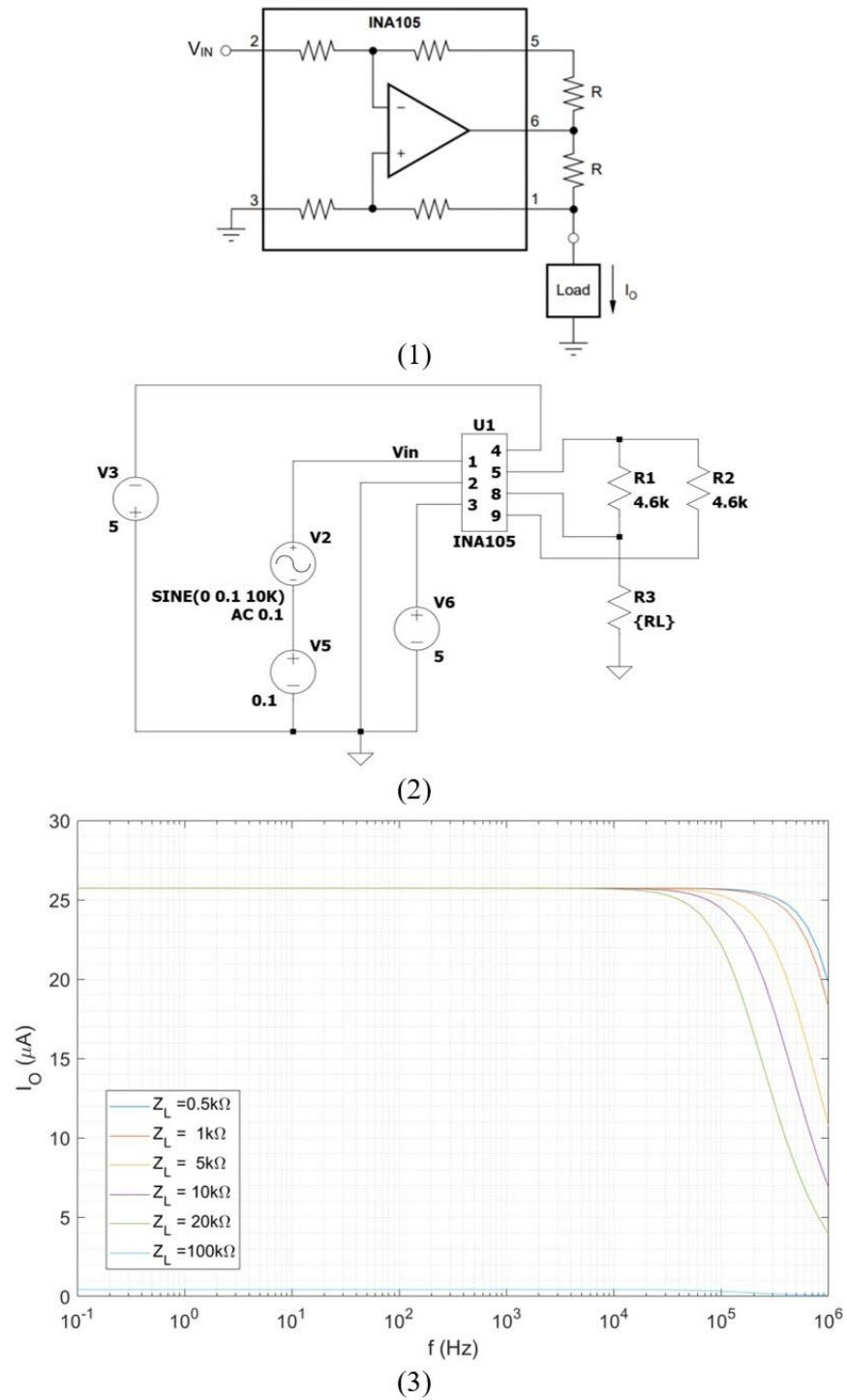


Figure 3.7. VCCS Design, Simulation, and Simulation Response. (1) VCCS Schematic as provided by INA105 documentation. (2) Complete circuit, simulated in LTSPICE XVII. (3) Simulated response of output current with respect to output load and frequency.

To determine the stability of our circuit at different input frequencies and output loads, we assume as previously mentioned that our frequencies will be less than or equal to 100kHz, and that we will remain less than 100k $\Omega$  of output load. The 100k $\Omega$  of output load is determined by assuming a tissue impedance range of roughly 1-10k $\Omega$ , thus 100k $\Omega$  should be sufficient margin [12].

Simulations were performed on LTSPICE XVII using the simulated integrated circuit and circuit component models to compare load values to output current. The schematic models used are depicted in Figure 3.6.2.

To verify expected operation of the full VCCS circuit, output current was plotted against variable output loads and frequencies, the results of which are shown in Figure 3.6.3.

We built the VCCS on a breadboard as pictured in Figure 3.7. To test the VCCS, we inputted a pulse from the RP of 0.1V and measured the output through the VCCS and output load resistors of 1k $\Omega$ , 5k $\Omega$ , 10k $\Omega$ , and 100k $\Omega$ . Input signal frequency was swept from 0.1Hz to 1MHz.



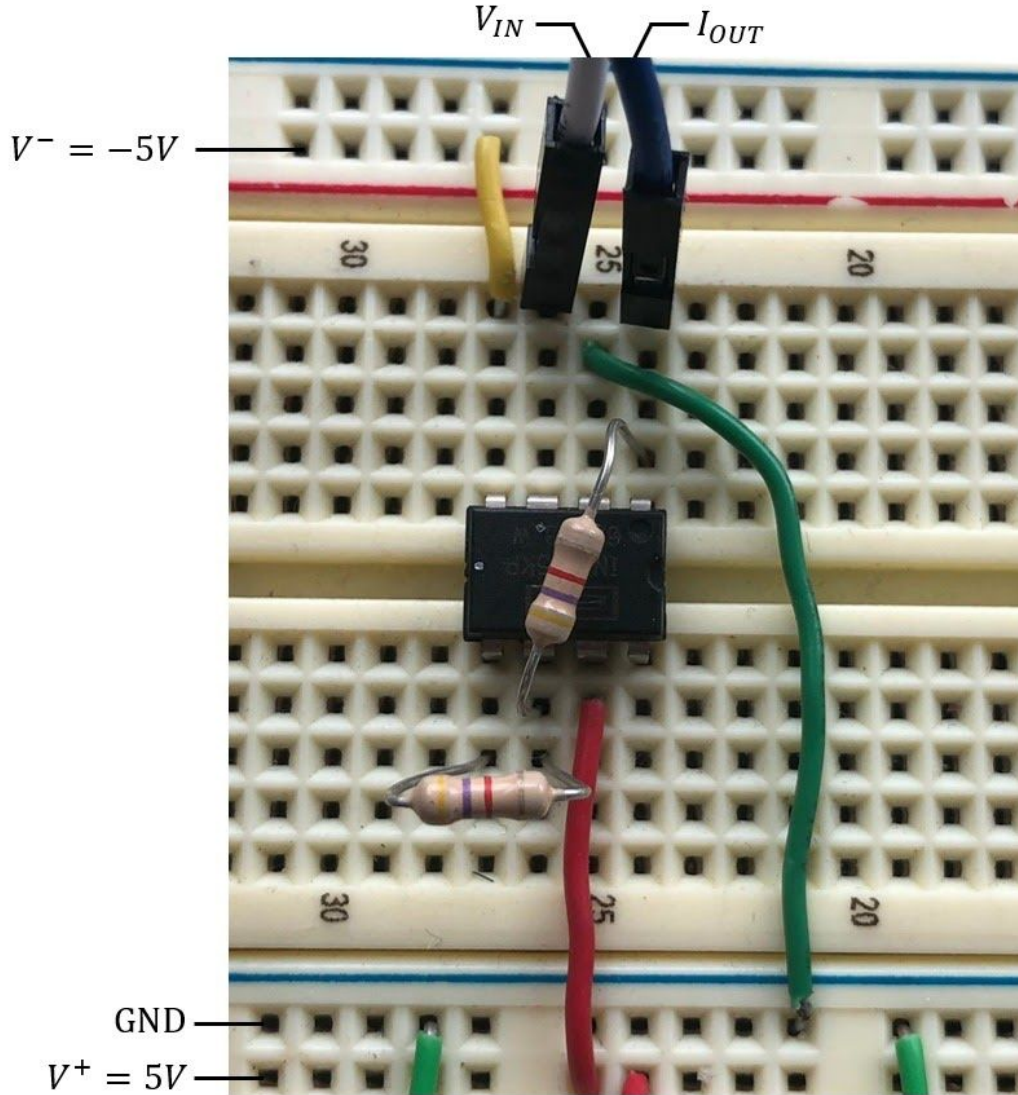


Figure 3.8. Photo of the VCCS Circuit Physical Construction

Output current as a function of input frequency and various loads is shown in Figure 3.8. To calculate the most relevant output impedance, we assumed the  $1\text{k}\Omega$  to  $10\text{k}\Omega$  range of biological tissues mentioned previously, and a maximum frequency of  $100\text{kHz}$ . Thus,

$$Z_O = \frac{V_1 - V_2}{\frac{V_1}{Z_{L1}} - \frac{V_2}{Z_{L2}}} \quad (16)$$

For a  $V_1$  and  $V_2$  of  $0.138\text{V}$  and  $1.312\text{V}$ , as measured across output load resistors of  $1\text{k}\Omega$  and  $10\text{k}\Omega$  respectively, we obtain an output impedance of  $142\text{k}\Omega$ , meeting our performance needs.

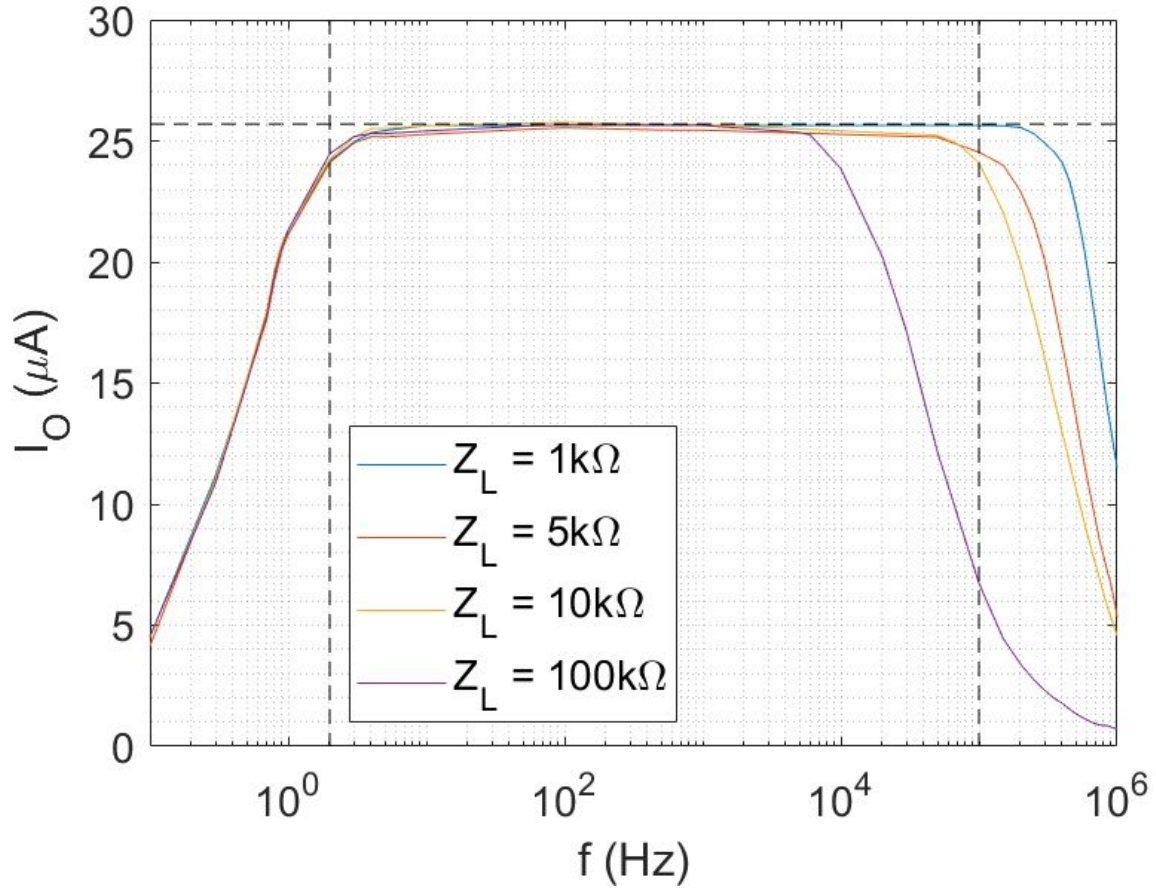


Figure 3.9. VCCS Circuit Output Current Response with Variable Frequency and Output Loads

### 3.4. Multiplexers

To expand the single channel output from the RP and VCCS, we need to quickly be able to direct the signal to the proper channel of the EIT tank, and to receive each signal properly as well. This is accomplished using an array of four individual 16-Channel, CD74HC4067 MUX boards from Sparkfun [28]. These MUX boards were chosen for their low impedance, low noise, and high switching speed.

16 channels allow us to expand the capabilities of previous portable systems by increasing resolution without any detriment to the system. These allow for a wide range of signals to quickly be selected digitally using digital addressing pins of the RP. Separating each signal (injection source, injection sink, voltage measurement, and ground) allows for the best performance, as it minimizes contact impedance [12], [29] and should minimize crosstalk from any capacitive coupling between the signals.

Biphasic signalling is also used to minimize contact impedance further and help prevent oxidation of some materials, and is useful clinically as it results in less tissue damage for a given amount of applied current [32].

We use a set of multiplexers to select each channel of the EIT tank as needed, whether it is set to current injection as the source, the injection current sink, voltage measurement, or ground. Each MUX board can take a signal and use digital addressing to output the signal to 16 channels. Each MUX is labelled as MUX1, MUX2, MUX3, and MUX4. The complete array is shown in Figure 3.9.

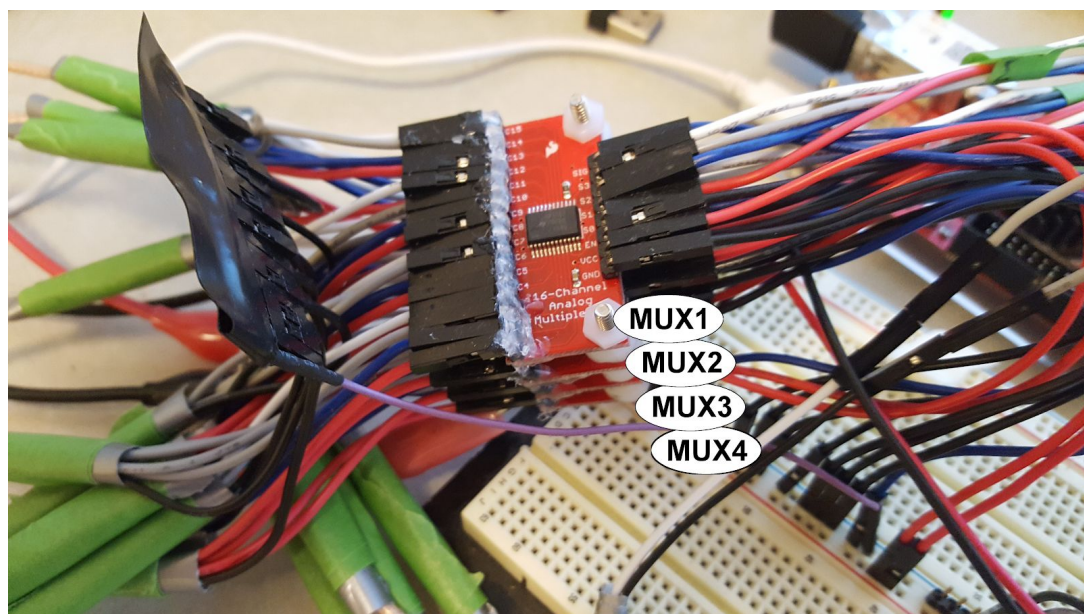


Figure 3.10. Photo of Multiplexer Array

MUX1 is connected and dedicated to the current injection source (the output of the VCCS). MUX2 is set as the current injection sink (circuit ground). MUX3 and MUX4 connect to the inputs of the amplifier and filtering circuits as described in section 3.1.5. All addresses are tabulated in Table 1.

Table 1. MUX Addressing Table

MUX	Addressing Pins				Common I/O	Function
	S0	S1	S2	S3		
<b>MUX1</b>	D0_N	D0_P	D1_N	D1_P	VCCS Output	Current injection output
<b>MUX2</b>	D2_N	D2_P	D3_N	D3_P	Ground	Current sink
<b>MUX3</b>	D4_N	D4_P	D5_N	D5_P	Non-inverting Input to IA	IA Input: measured signal
<b>MUX4</b>	D6_N	D6_P	D7_N	D7_P	Inverting Input to IA	IA Input: reference value

Power is provided through the same 5V power source as the RP and other circuitry. Although 64 individual pins had to be connected manually, the MUX boards do remain modular as individual connectors can be disconnected or moved as needed.

### 3.5. Tanks and Cables

All tanks and electrode cables were custom-built. Tanks were designed in Blender and 3D-printed using a Form 2 high-precision stereolithography resin printer [23],[30]. Both tanks were designed to be virtually identical. Dimensions of the cylindrical tanks are 112mm diameter by a height of 25mm. 16 channels of electrodes are aligned equidistant from one another, centered at a height of 12.5mm from the bottom of the tank. Indentations were made to make object placement easier and more standardized. Tank objects were also 3D printed, and the object used in this project's analysis was specifically a 50mm diameter cylinder.

The cables were soldered by hand, with all shielding grounded through the multiplexer boards.

#### 3.5.1. Point-Electrode Tank

The point-electrode tank was made using female SMA RF connectors that terminated in gold-plated needle-like protrusions on the electrode side. The SMA connection allowed for a low-noise connection to the cables, which allowed for the tank to be removed in a modular fashion. Gold was preferred as a contact material due to its corrosion-resistant properties, and therefore better signal transmission [31]. Electrodes were sealed with silicone adhesive. A photo of the point-electrode tank is given in Figure 3.10.

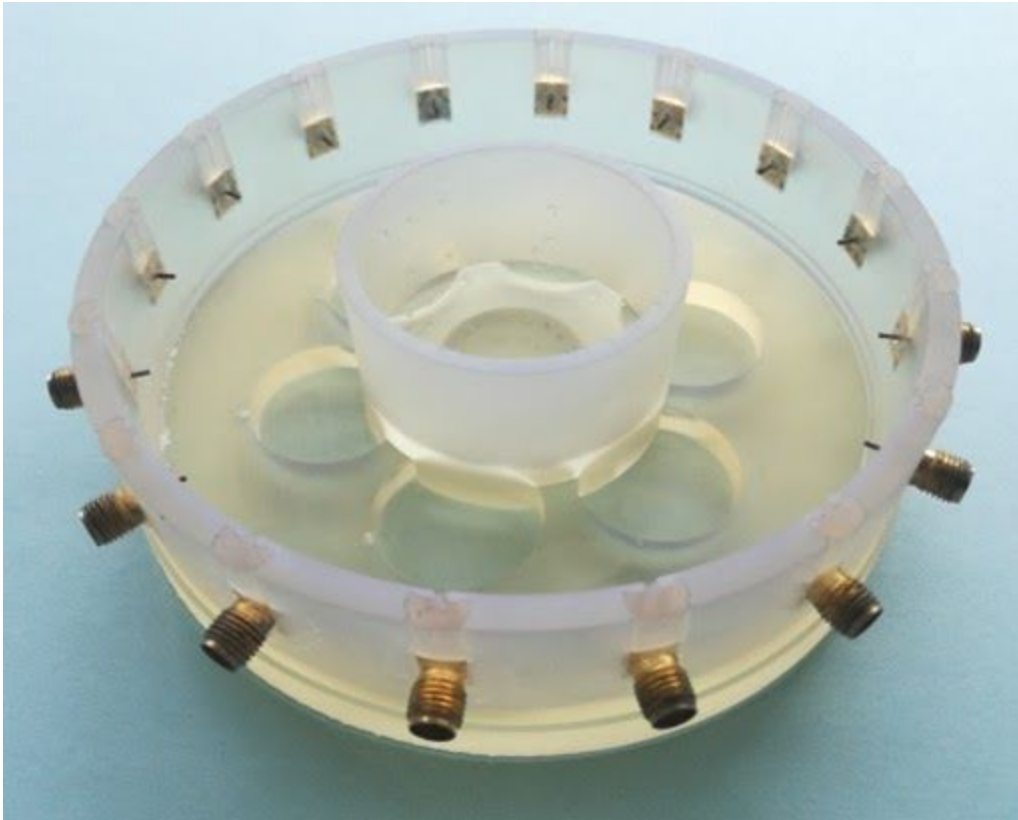


Figure 3.11. Point-Electrode Tank, with Object

### 3.5.2. Plate-Electrode Tank

The plate-electrode tank used similar SMA RF connectors, however these contacts were soldered to 11mm by 11mm square copper plates, with 10mm by 10mm of area exposed to the tank. Electrodes were sealed with epoxy and silicone adhesive. A photo of the plate electrode tank is provided by Figure 3.11.



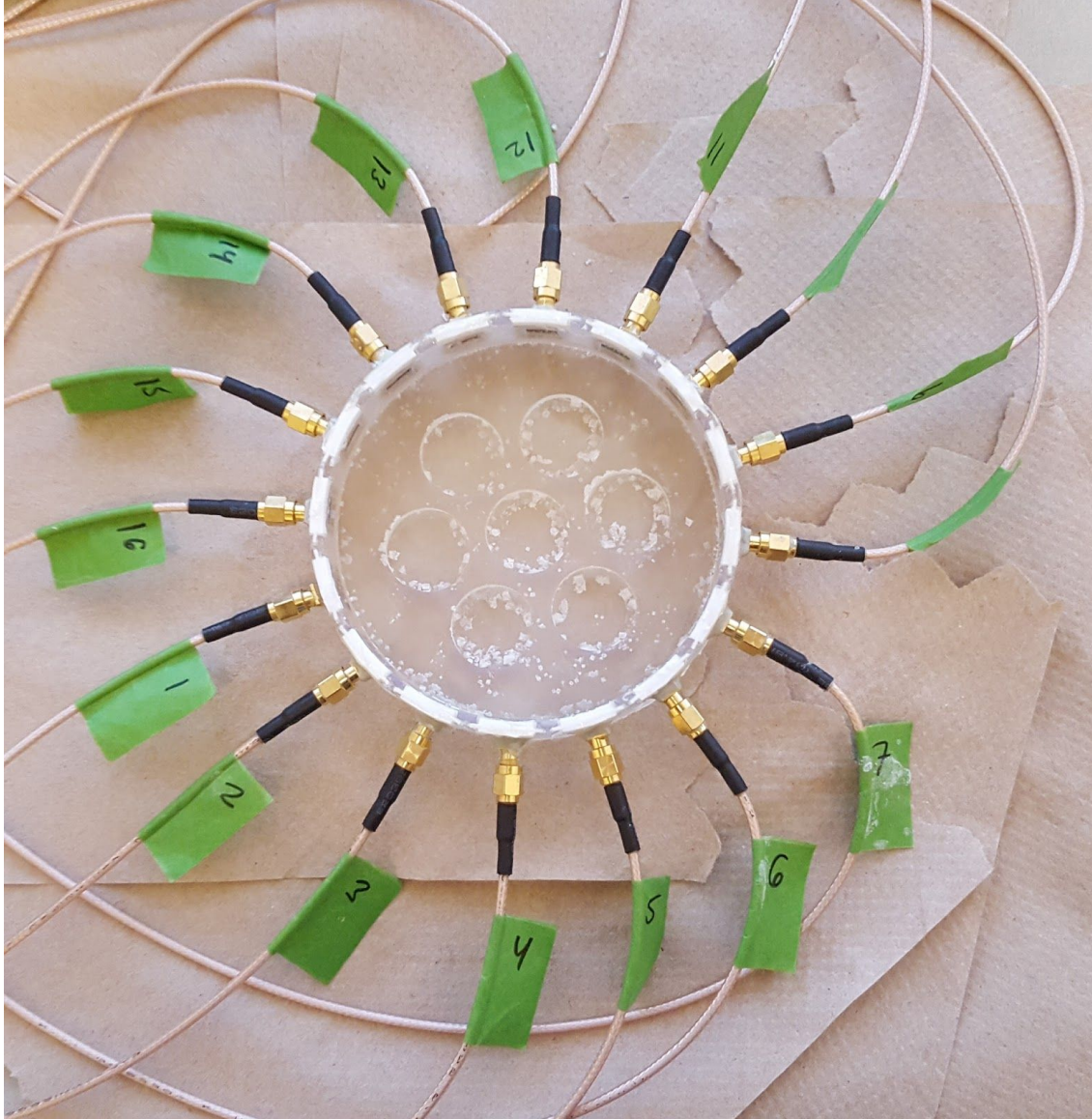


Figure 3.12. Plate-Electrode Tank

### 3.5.3. Cables

Cables were custom-built to connect from the MUX array to the EIT tank. Copper-shielded cables with gold-plated SMA RF male connectors were split in half, and each open end was soldered to female pin header connectors. Each cable connects one EIT tank channel to all four MUX boards at that specific channel, as well as connecting the shielding to ground of the MUX boards. A picture of one of the cables made is shown in Figure 3.12.

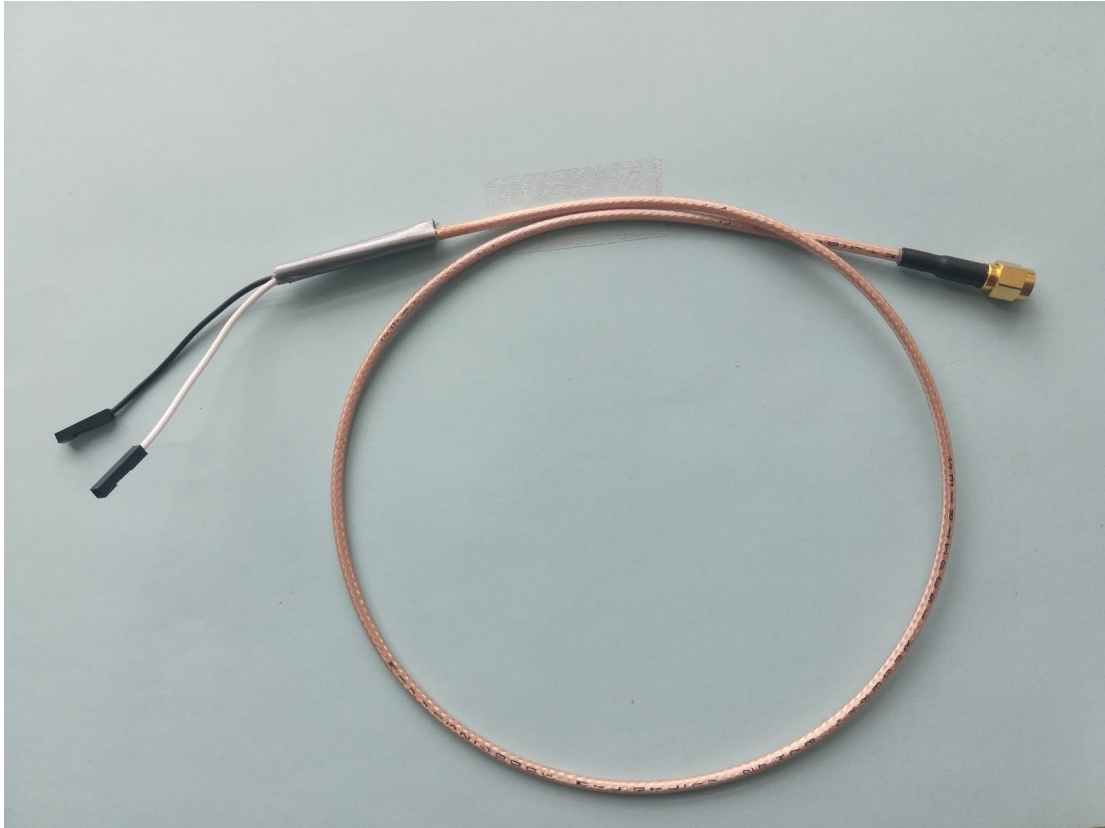


Figure 3.13. Custom Cables for EIT Tanks

### 3.6. Amplifiers and Filters

To post-process the signals from the EIT tank (and through the MUX's), we must amplify and filter the signals. To do this, we need a multistage device which consists of multiple individual circuits joined together. The whole circuit needs to have a very high input impedance to resist drawing of current from the EIT tank during the data acquisition process, and to maintain the operating bandwidth of the VCCS circuit and the total EIT system (100kHz), as determined previously. Noise should be minimized as much as possible by matching impedances between subsystems of the full circuit [33], and lastly, the output voltage should remain less than 1V so as to remain in full range for the RP input. Individual subcircuits consist of an instrumentation amplifier, a DC removal circuit, a band-pass filter, and single-ended output amplifier.

The first stage is made using an AD620 Instrumentation Amplifier (IA) [34]. This provides the high input impedance to the amplifier and filter circuits, with low noise and a 1MHz bandwidth (exceeding our

needs). This amplifier is also notable for its low Common Mode Rejection Ratio (CMMR). CMMR is a measure of an amplifier's ability to reject noise that is common to both inputs [35]. Thus, a high (ideally infinite) CMMR denotes an amplifier with minimal common-mode noise, and is a highly desirable characteristic of our input amplifier. To select the gain, we simply need to select for  $R_G$ . By setting  $R_G = 10\text{k}\Omega$ , we receive a gain of 5.94 from the equation:

$$R_G = \frac{49.4\text{k}\Omega}{G-1}, \quad (17)$$

At this point, we have amplified the signal, but still need to remove the DC components necessary for the MUX array to consistently stay above 0V, and therefore operational. This is achieved through a DC-removing circuit, designed with a LM741 OPAMP and coupled directly to the output of the IA. To this end, we create what is known as an integrator circuit. This is used in lieu of an AC filter prior to the IA, which would have drastically altered the system response. This integrator simply subtracts the observed DC signal from the output of the IA using a capacitive feedback loop, and leaving the AC signal intact. This combination of IA and DC-removal circuit can be seen in Figure 3.13.



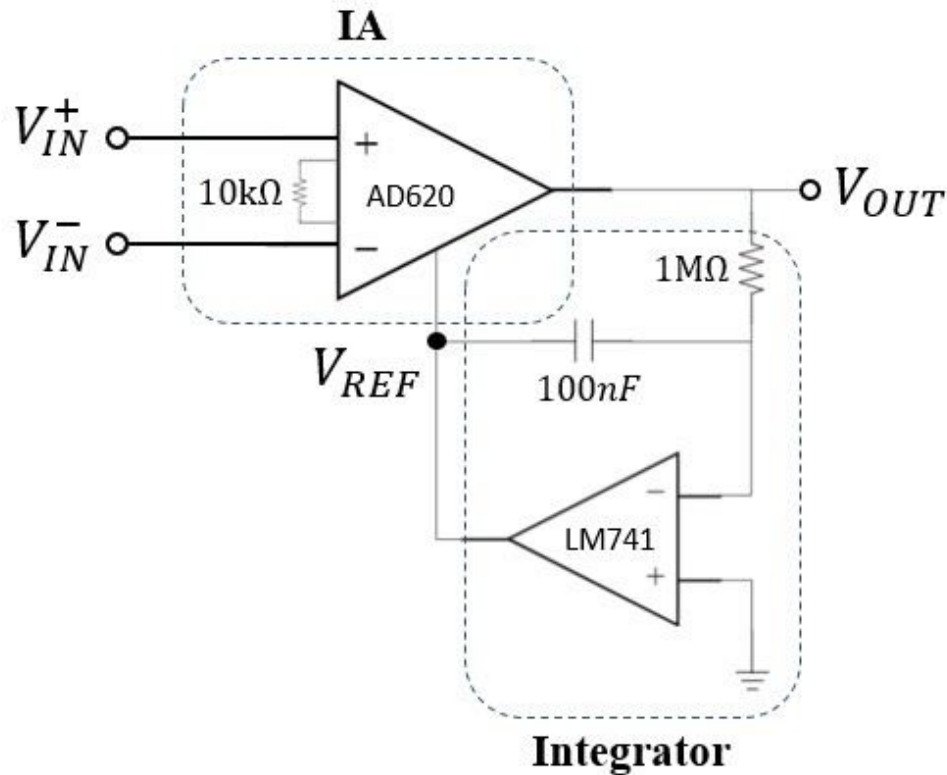


Figure 3.14. Instrumentation Amplifier with Attached DC-Removing Integrator Circuit

From here, we can begin to filter the signal using simple AC circuit design. We desire to filter out all DC signals, and all signals above 100KHz. Sources of low-frequency noise often include environmental noise, such as background AC source voltage [36], while high frequency noise is more likely to occur from nearby electronics or from the circuit itself [37]. A band-pass filter is constructed from an in-series high-pass and low-pass filter, leaving a functional band of 0.7Hz to 117kHz. Any noise below 0.7Hz AC, including DC, is removed, as well as any possible noise outside our bandwidth.

Lastly, an output amplifier is needed to amplify the filtered signal. To this, we use a single-ended amplifier as our output amp, for a direct connection into the RP without any other signal interference. Amplifying the signal after filtering allows for the maximum value of processed signal to reach the RP. This is done using an LM741 OPAMP as in the DC removal circuit. In this circuit, we refer to the amplifier documentation provided by Texas Instruments [38], with a schematic template shown in Figure 3.14.

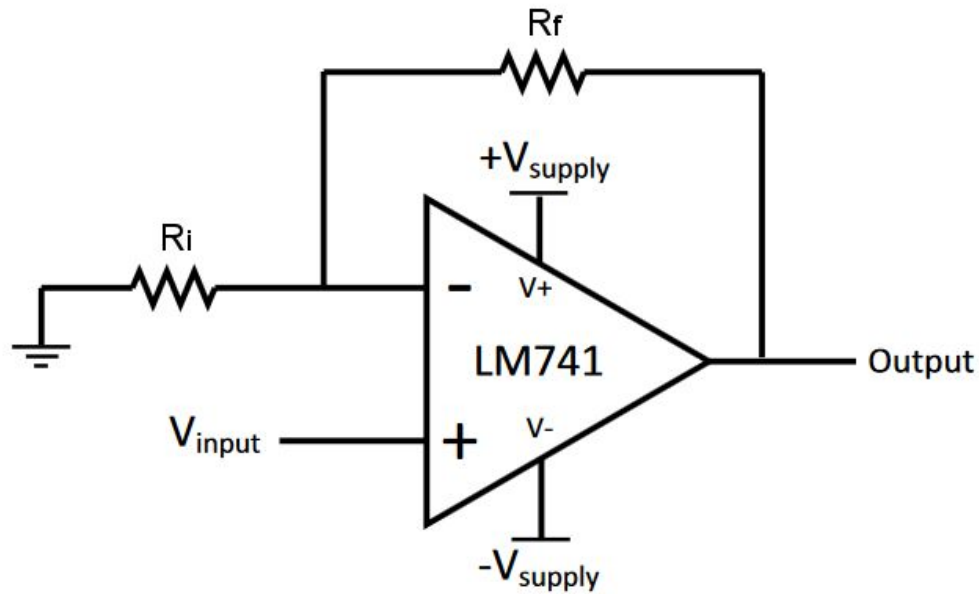


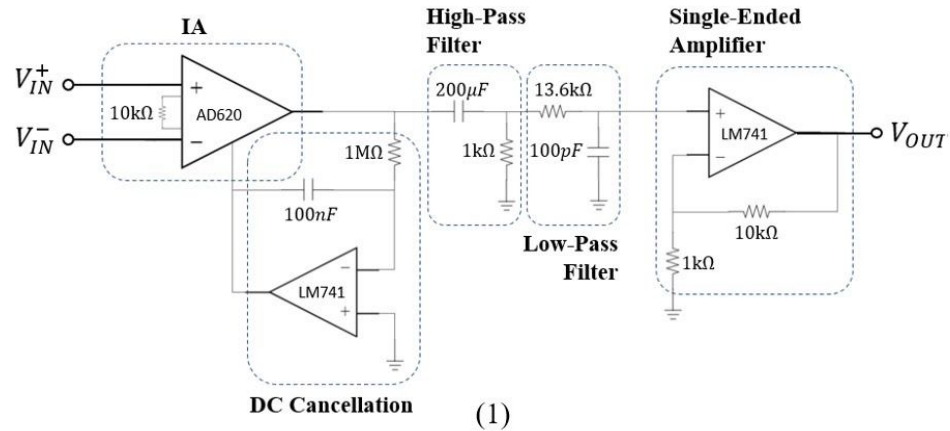
Figure 3.15. Schematic Template Used for Output Amplifier

To find the total Gain, we refer to the equation from the datasheet:

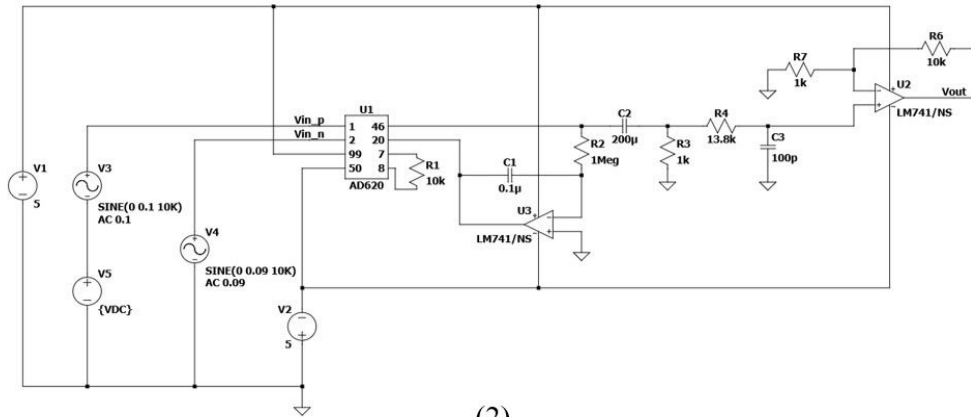
$$Gain = 1 + \frac{R_f}{R_i}, \quad (18)$$

We set  $R_f = 10k\Omega$ , and  $R_i = 1k\Omega$ , for a gain of 11. This is adjustable through resistors  $R_f$ , and  $R_i$ , however we keep these values constant so as to ensure the correct system response throughout all conditions. Combined with the previous gain of 6, total circuit gain = 66. This equates to a predicted decibel gain of  $20 \log (Gain) = 36.39 \text{ dB}$ .

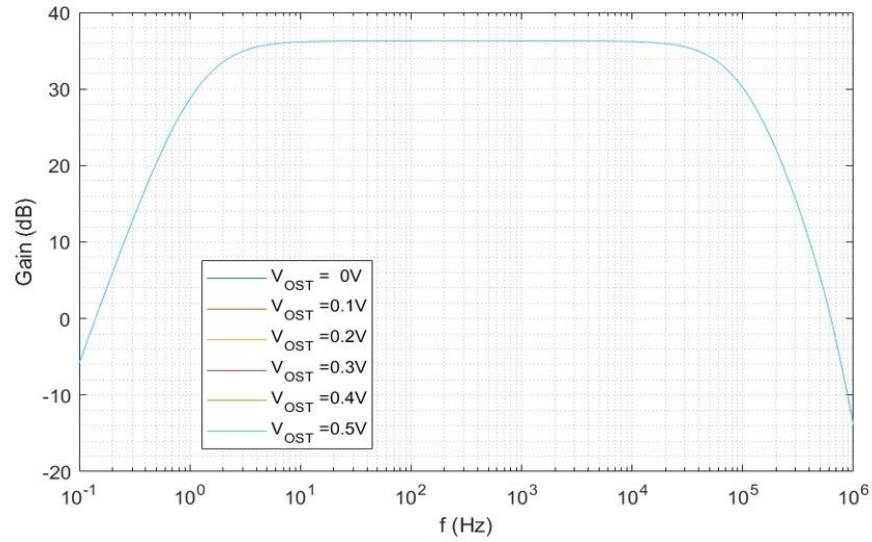
The total combined circuit was simulated on LTSPICE XVII. Simulated input voltages were AC signals of  $V_{IN}^+ = 10\text{mV}$  and  $V_{IN}^- = 9\text{mV}$ , for an increasing DC offset. Simulated system gain response is shown in Figure 3.15.3. The DC offset of the input shows no change in output gain. Frequency response drops off near 50KHz (only half of the previously described 100kHz upper limit), but maintains a 30dB gain at a 100kHz signal input.



(1)



(2)



(3)

Figure 3.16. Schematics of Amplifier, with Testing. (1) All amplifier and filter circuits are connected together and shown as distinct, interconnected parts. (2) All circuits are simulated using LTSPICE XVII, and (3) the simulated circuits are tested to ensure adequate response for the system.

All circuits were physically constructed on a breadboard, and are labelled clearly in Figure 3.16.

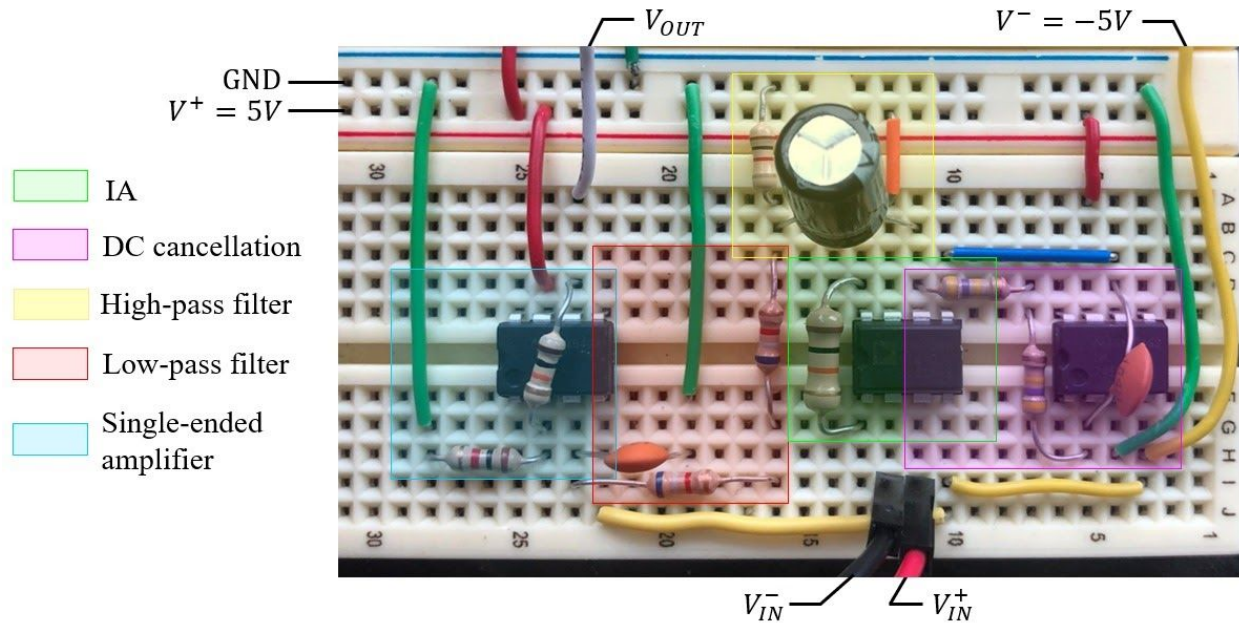


Figure 3.17. Photo of All Amplifiers and Filters, Labelled

Similar to the VCCS circuit, we test the response of the physical circuit to ensure proper operation. The test response to the physical amplification and filtering circuits can be seen in Figure 3.17. Bandwidth is roughly 100kHz, and gain is about 36dB for a majority of bandwidth.

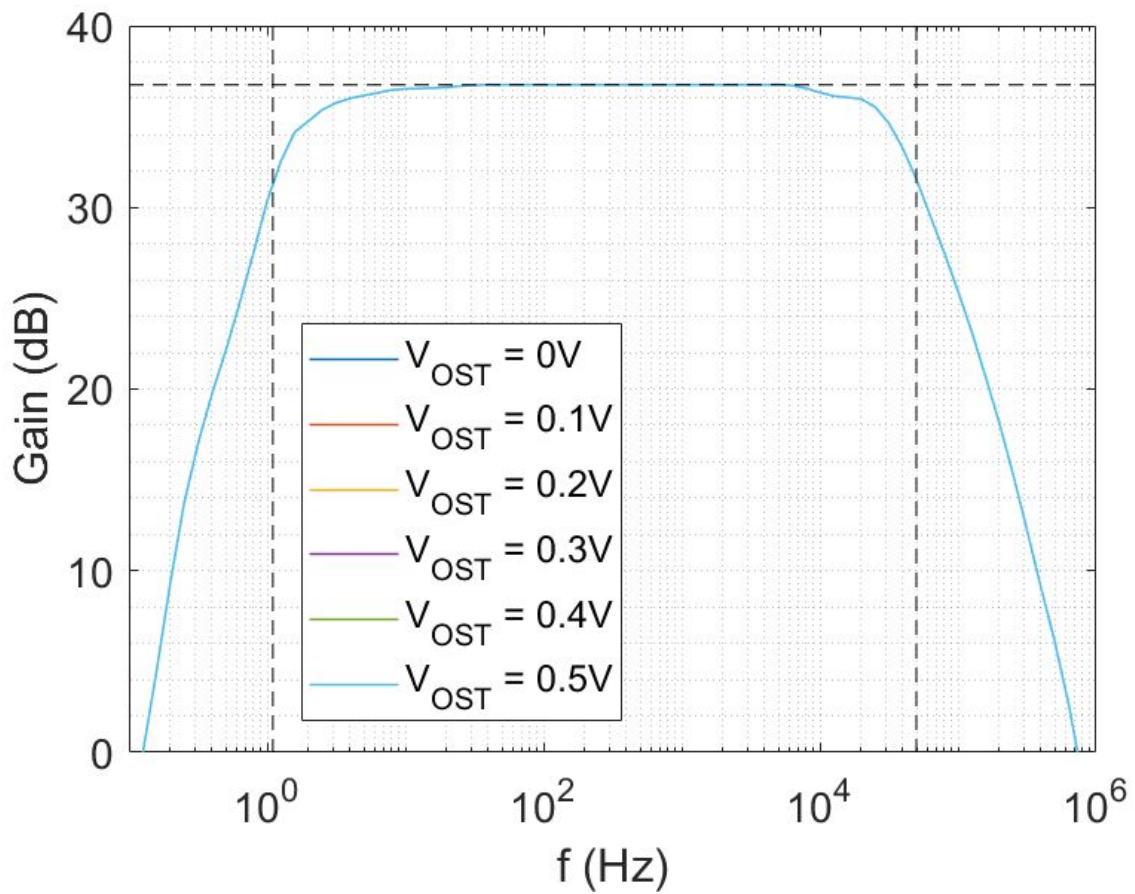


Figure 3.18. Results of Output Amplifier Across Various Frequencies and DC Offsets

All signal processing circuits appear together on the breadboard in Figure 3.18, and a full-system schematic is shown in Figure 3.19.



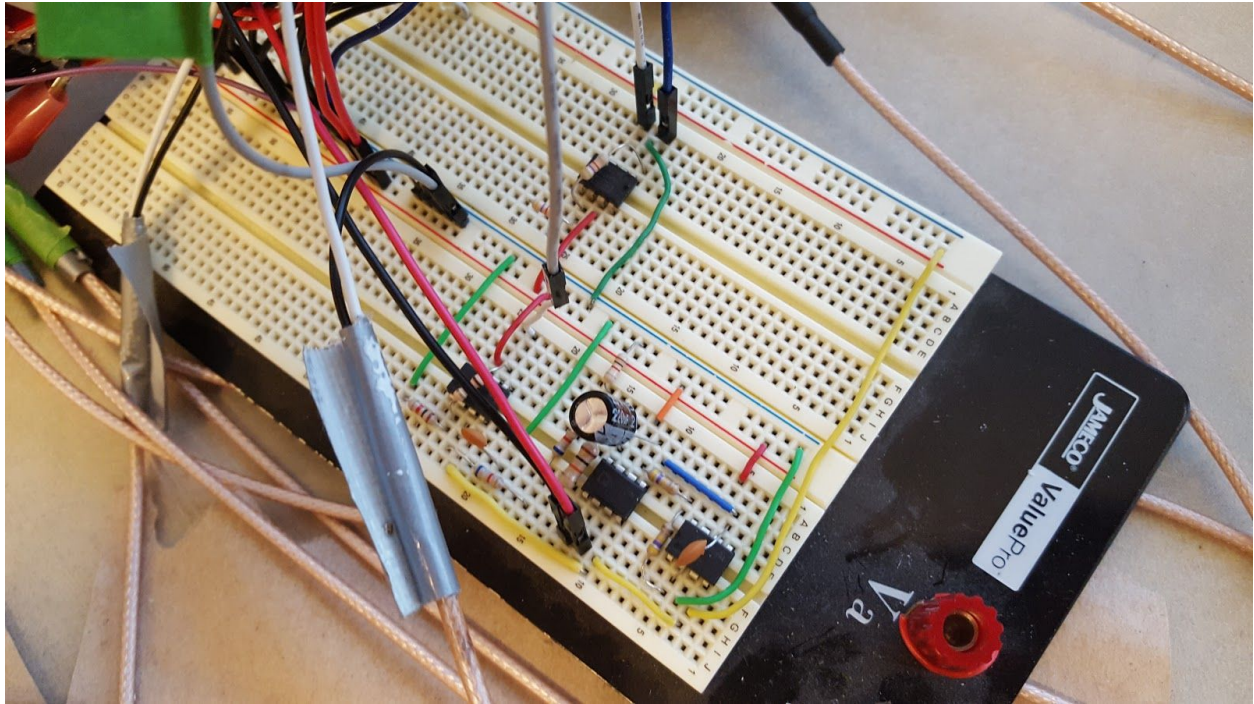


Figure 3.19. Photo of All Signal Circuits Together

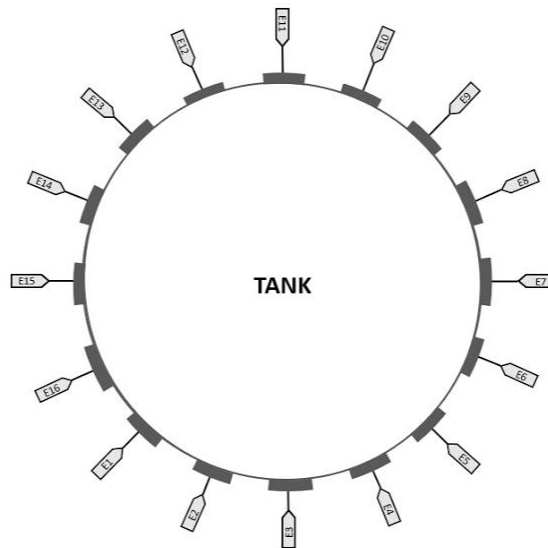
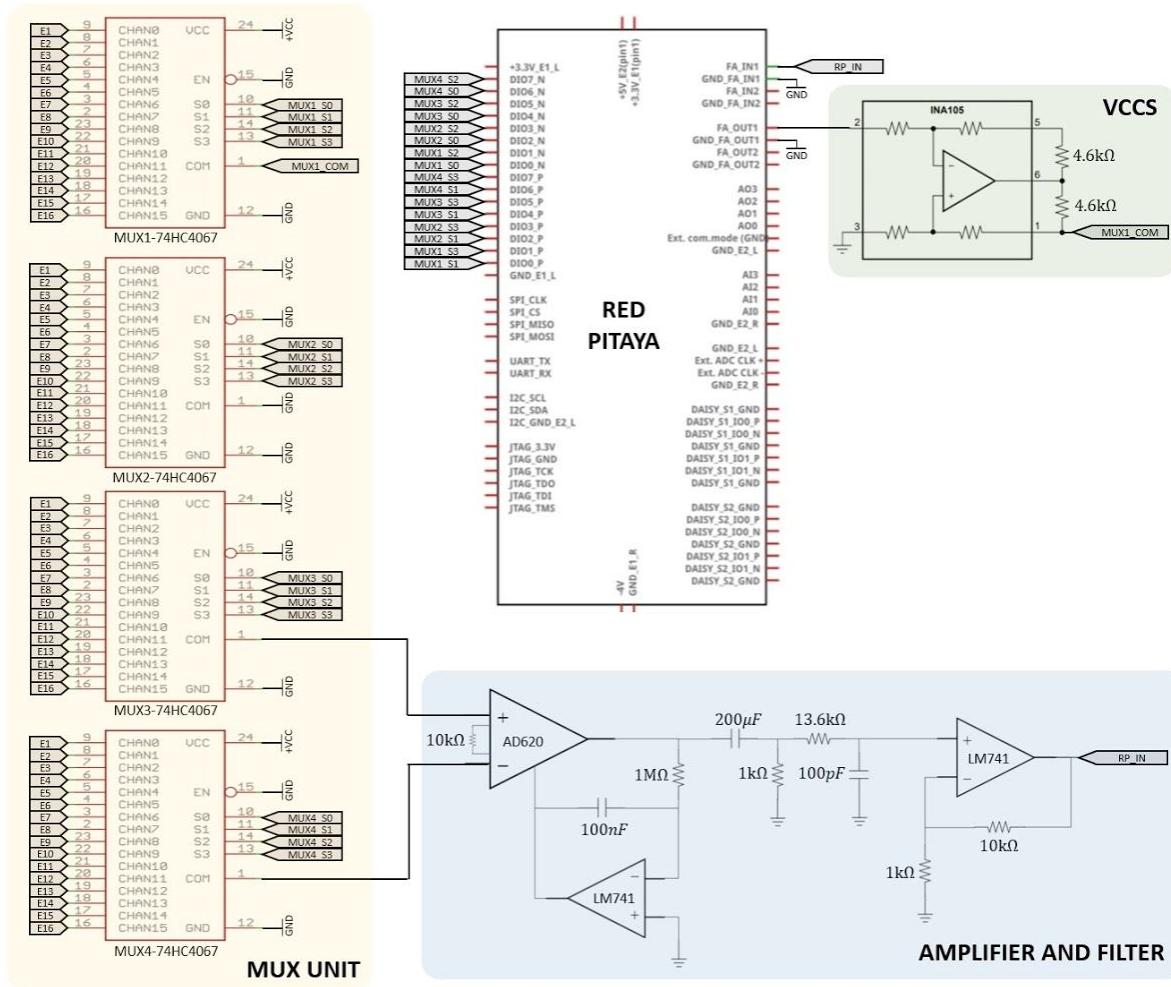


Figure 3.20. Full System Schematic

An example of other possible object configurations is shown in Figure 3.20, using two small objects. The image corresponding to this type of tank configuration is shown in Figure 3.21, including the unusual artifacting that multiple small objects appear to lead to. To clarify, this configuration is not quantified in our Methods or Results.

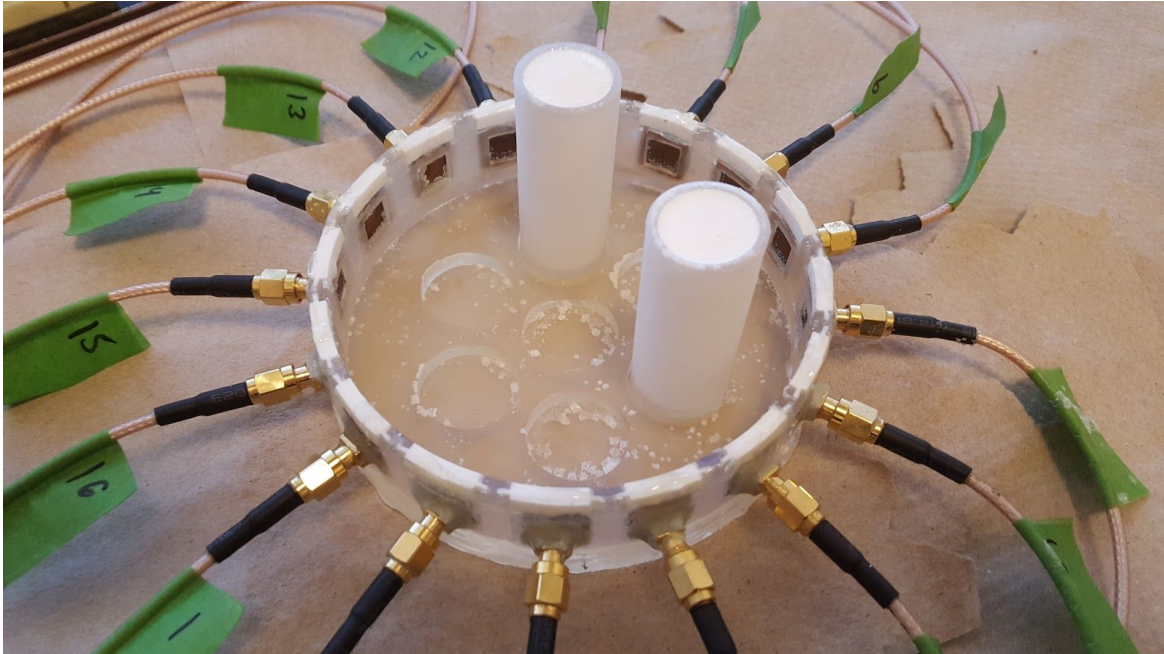


Figure 3.21. Photo of Plate-Electrode Tank, with 2 Objects

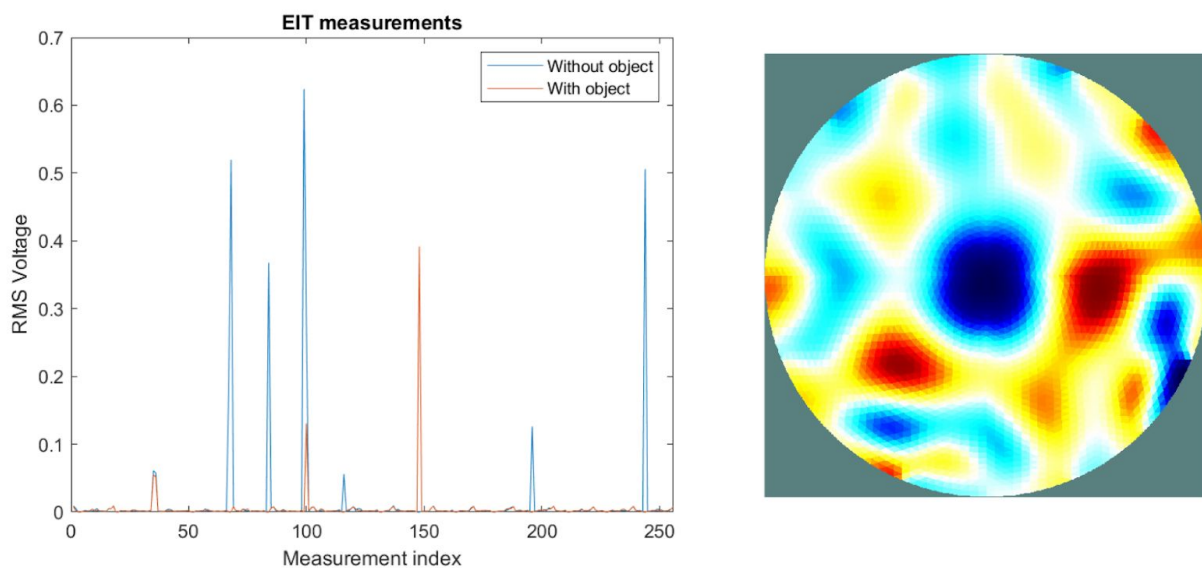


Figure 3.22. Example Data to Reconstructed Image



## **4. Software**

### **4.1. Software Overview**

Software used in this project includes the software of the RP itself, MATLAB (on a connected PC), and the MATLAB software set known as the Electrical Impedance Tomography and Diffuse Optical Tomography Reconstruction Software (EIDORS) [39].

To connect to the RP, the RP is configured to act as a wi-fi access point. The RP software is able to be accessed through this access point via a web browser on a connected computer. In its development section, a Standard Commands for Programmable Instrumentation (SCPI) bridge is set up, and can be connected and interfaced through MATLAB.

The process of gathering the data and reconstructing the images are performed by two separate MATLAB scripts, named “hardware\_control.m,” and “image\_reconstruction.m,” respectively. The first script controls the data acquisition process, and the second script takes the acquired data and uses EIDORS to reconstruct the images. The data acquisition process is shown in Figure 4.1. Code is included in the Appendix.

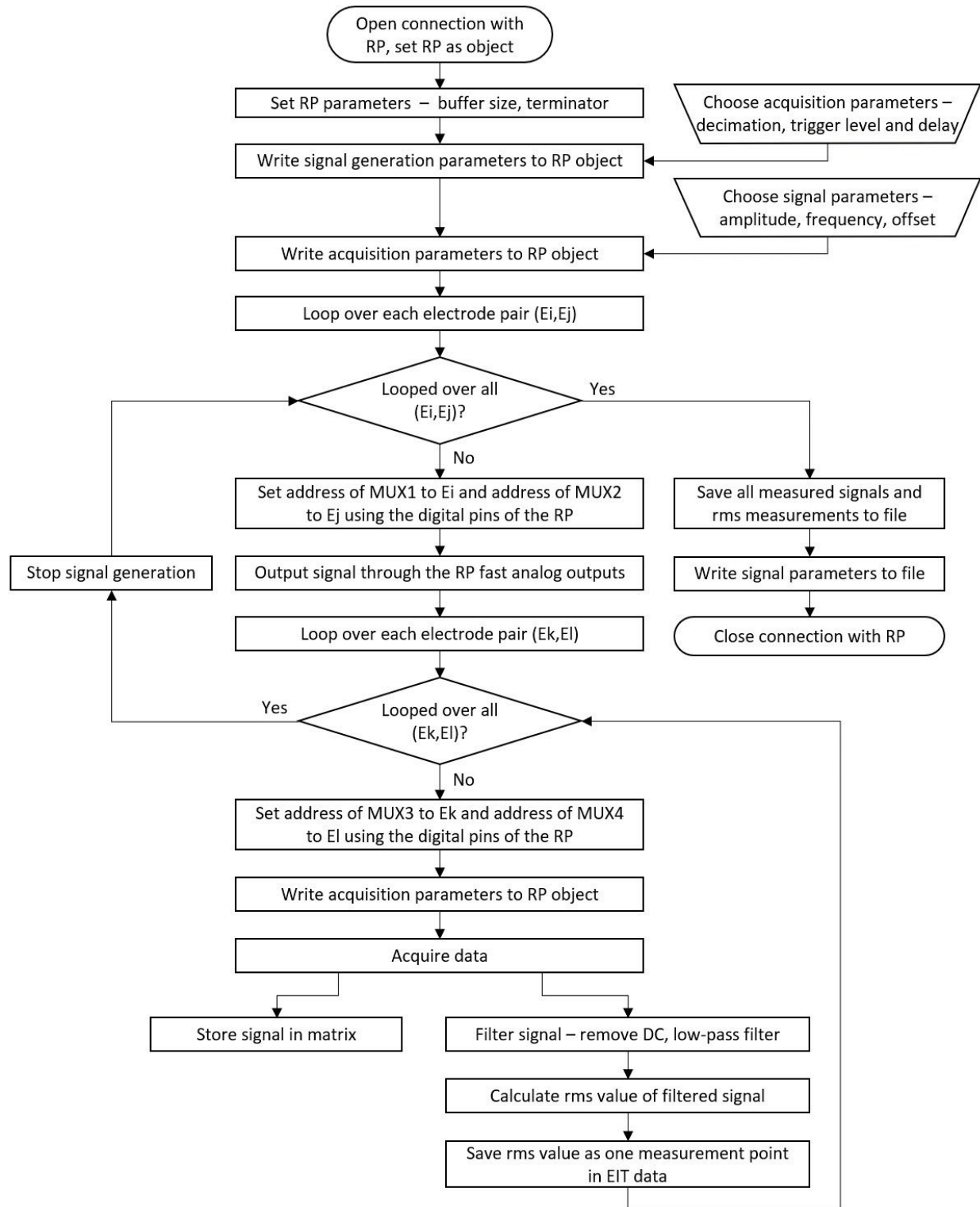


Figure 4.1. Data Acquisition Flowchart

## 4.2. Data Acquisition

In the MATLAB script “hardware\_control.m,” the RP is connected through a TCP/IP connection using the RP’s IP address, and through the use of SCPI commands. Once the connection between the PC running MATLAB and the RP is confirmed, initialization of the injection signal begins. This includes setting up the biphasic pulse frequency and amplitude. The biphasic pulse helps to prevent a net charge from being injected into the tissue, while also keeping a response that is frequency-dependent and yet ideally consistent with regards to the length of time of the actual pulse [32], [40].

Next, the parameters are initialized for the acquisition of the signals to create the data. These include the trigger level and delay, and also the decimation factor. From here, iteration for the injections can begin.

For each pair of adjacent electrodes, one pair (starting with electrodes 1 and 2), is selected by MUX’s 1 and 2 to act as the injection pair. The first electrode is channeled through the MUX array to the injection signal from the VCCS, while the other electrode acts as a current sink. Together, they form the current injection path. Meanwhile, all other electrodes are selected by MUX’s 3 and 4 to act as measuring points, starting with electrode 3. The signal is pulsed from the RP into the VCCS, and the current is injected from electrode 1 to electrode 2, while the other electrodes can measure the current, and feed this information back through the MUX array and into the filtering and amplification circuitry, and eventually back to the RP, where the data is sent to the connected PC and stored in MATLAB as a variable.

With each injection between the first two electrodes, MUX’s 3 and 4 continually increment the measuring electrode, from electrodes 3 to 16. Once each electrode has gathered the measurements from the first set of injections, the injection pair iteration begins to increment, from electrodes 1 to 2, to electrodes 2 to 3, and so on, until all adjacent pairs of electrodes have injected current, with measurements gathered by all of the other electrodes.

Each collected measurement consists of roughly 16,000 data points, from which a root mean square (RMS) voltage value is calculated. In total, this creates 256 individual voltage values (boundary values) through which we can begin image reconstruction.

#### **4.3. Image Reconstruction**

In the script “Image\_Reconstruction.m,” EIDORS is used as the main toolkit for image reconstruction. It allows for the creation of both forward and reverse Finite Element (FE) models, as well as the reconstruction protocols to create image approximation from the models.

First, two different sets of data must be loaded into MATLAB. Each data set is made of the 256 voltages, acting mathematically as the boundary values. Because images will be constructed with differential imaging, an “empty” tank of data must be used for a reference. This allows for many of the potential artifacts to be reduced, but it also comes at the cost of requiring an “empty” image, which may not be practical in medical applications. That is, it may not be feasible to have “reference” tissues or structures available to compare to in a clinical setting. Still, this allows for reduction of artifacts and is practical in our case.

Once the data is loaded in, some measurements are removed from calculation. The injecting electrodes are not considered in calculations at all (as they are not actively measuring voltages), and electrodes adjacent to the current-injecting electrode are removed to minimize noise. For each of the 16 possible electrodes from which data can be acquired, per injection cycle, this leaves 13 data points. In total, this gives us 208 data points to reconstruct from.

Next, a FE forward model is created from the data. This then allows for the inverse to be solved, so as to create the state of the image which would produce the boundary values measured.

To solve this for the forward model, we employ a one-step Gaussian-Newton reconstruction algorithm, which results in an approximation of the initial state of the tank and all objects within, corresponding to the measured voltages relative to the injected current. This end result is our acquired image.

In order to use the one-step Gaussian-Newton reconstruction algorithm, a “hyperparameter” is required. In our case, we use a “k-value” that we determined through observation to best visualize the observed objects. In the point-electrode tank, a k-value of 0.01 was used. For the plate-electrode tank, a k-value of 0.001 was used. This led to increased artifacts, but was needed to capture tank objects. Various k-values are shown side by side in Figure 4.2., illustrating the importance of finding the optimal k-value.

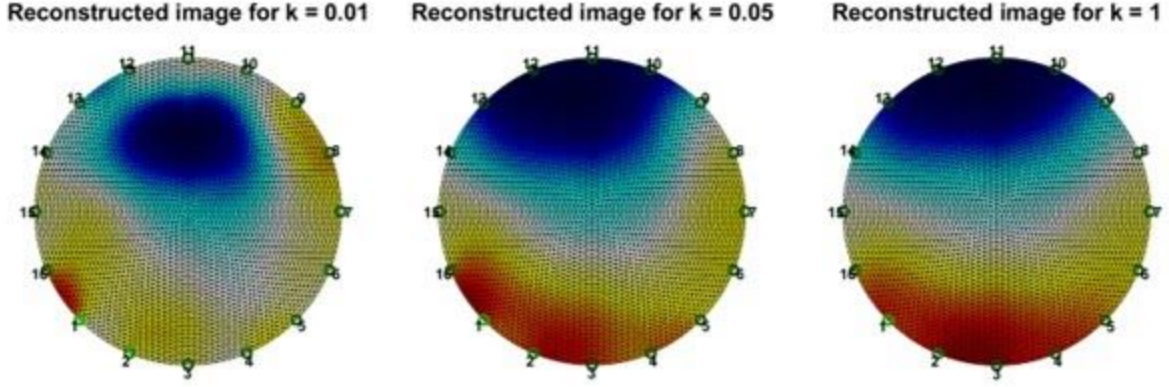


Figure 4.2. Various K-Values used in Reconstructing Images with the Point-Electrode Tank

## 5. Methods

The ultimate results of the system are quantified according to the Signal-to-Noise Ratio (SNR) of the final images, as well as the accuracy of the final images relative to corresponding simulated images. SNR is an often-used metric for quantifying medical imaging modalities, and it allows for both an intuitive understanding of the quality of the final images, and for comparison to other modalities. SNR compares the quality of the measured signal to that of the background noise and is typically measured in decibels (dB). Accuracy is determined by the use of a 2D Pearson coefficient ( $r$ ) and by analyzing characteristics of the objects after reconstruction and thresholding.

SNR is quantified in two steps. First, each channel is quantified individually. To do this, electrode pairs (1, 2), (7, 8), (12, 13), and (15, 16) were used for injection current signals, while the other electrodes were measured. As with the process of removing voltage measurements by adjacent channels, we discard these to remove noise. We inject signals over a hundred times per channel, and compute the mean and standard deviation of the voltage values. In total, 200 measurements were used in this analysis.

We use measures of voltage to calculate our SNR, which is an amplitude and not a measure of power. Therefore, to make it meaningful relative to power, SNR is calculated as follows [41]:

$$SNR_{channel (dB)} = 20 \log_{10} \left( \frac{A_{signal}}{A_{noise}} \right) = 20 \log_{10} \left( \frac{\bar{V}}{\sigma_V} \right), \quad (19)$$

for  $\bar{V}$  and  $\sigma_V$  as the mean and standard deviations of the voltages measured, respectively.

Additionally, we provide SNR values after filtering and prior to reconstruction. We also provide the system's total SNR as the mean value of all channel SNRs, as calculated above. SNR data was gathered for both filtered and unfiltered EIT tank signals.

In order to quantify the accuracy of the reconstructed images, we begin by capturing image data of objects in the tank, and by comparing the reconstructed images to idealized simulations. Code for this process is found in the Appendix. These simulations use FE models designed to perfectly simulate an object of high conductivity at any point determined by the user. Objects in the real world are placed with the nearest edge 15 mm away from each electrode, and simulated objects are placed accordingly, though purely within MATLAB, and reconstructed as though they were real-world objects. This provides a comparison of real objects to ideal objects. Figure 5.1 shows 8 of the 16 positions of the objects, using the point-electrode tank. Figure 5.2 shows the corresponding conductivity maps of each position.

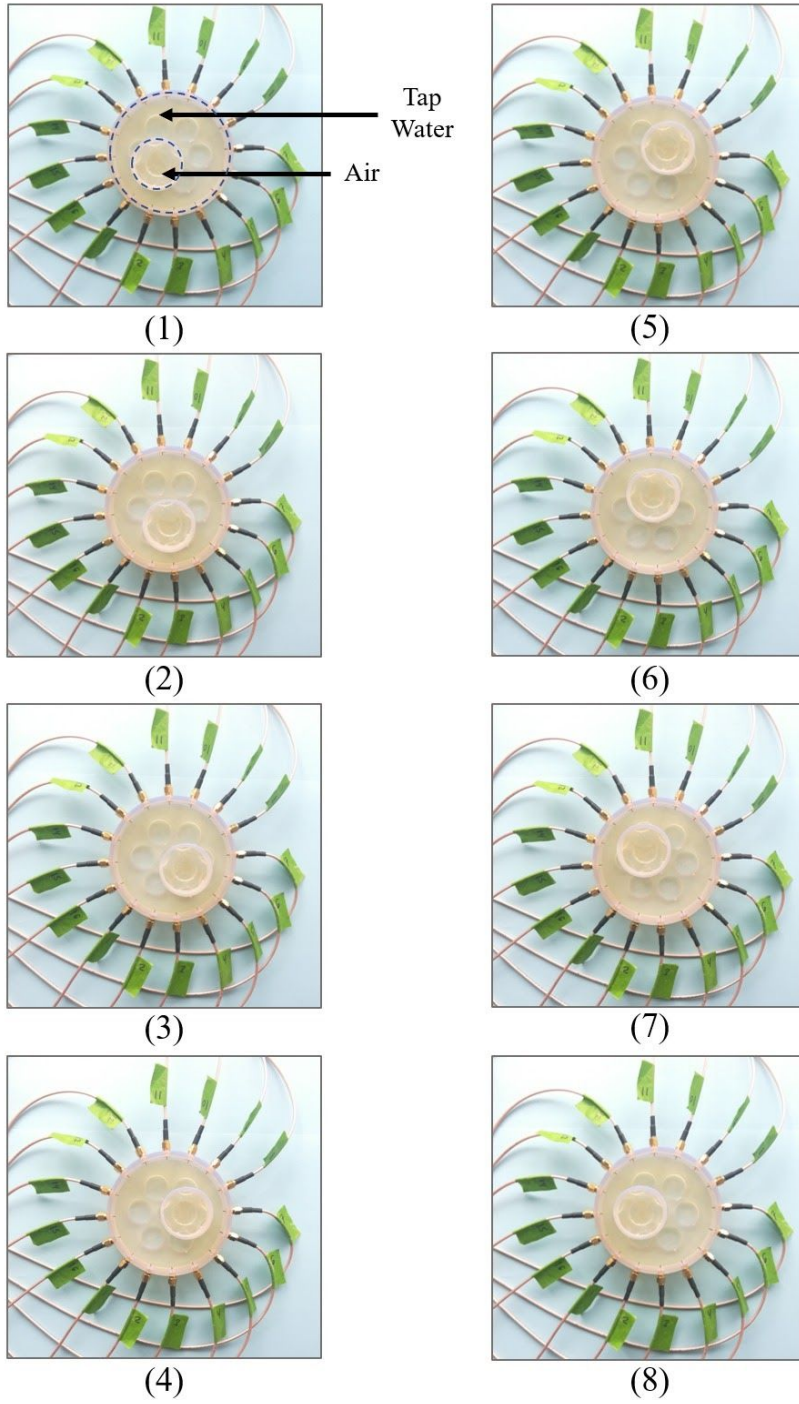


Figure 5.1. Testing the point-electrode tank object reconstruction accuracy in 8 example positions. The plate-electrode tank was tested with the same configurations. Positions 1 through 8 shown correspond to 15mm distance from electrodes 1, 3, 5, 7, 9, 11, 13, and 15, respectively.



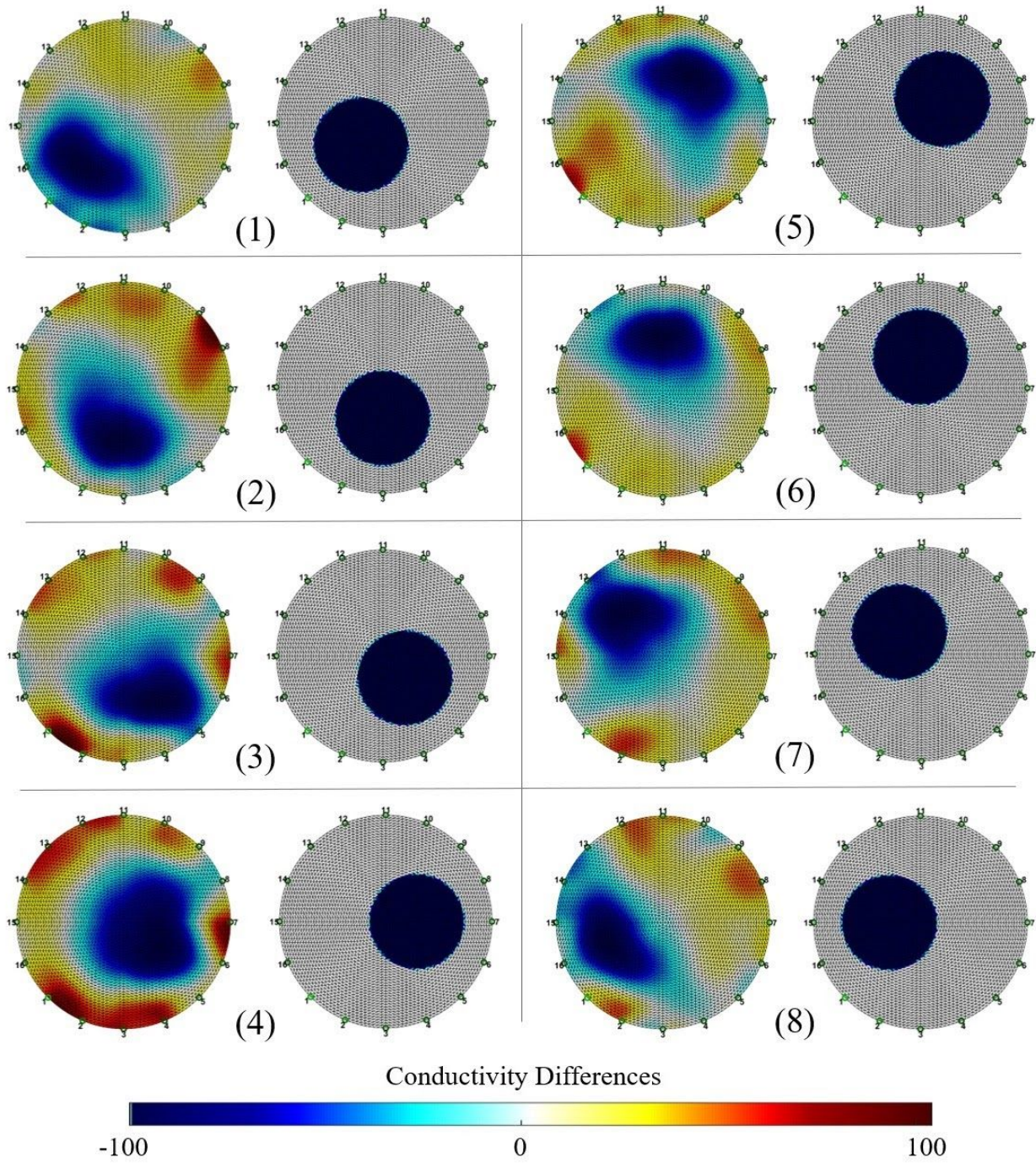


Figure 5.2. Point-electrode tank object reconstruction in 8 example positions (left) with the equivalent simulated object (right). The plate-electrode tank was tested with the same configurations with negative results. Units are absolute (no conductivity observed, to infinite conductivity observed).



The 2D Pearson coefficient, referred to simply as “r,” plots two variables against one another to show the linear relationship between the two. It also reduces the relationship to a single, meaningful number that allows comparison to other variables similarly. Both magnitude and direction of the relationship are provided by the number. This is also one of the most common metrics used for direct comparison, maximizing its meaningfulness to readers not necessarily familiar with EIT or our system. The 2D Pearson coefficient is computed in MATLAB as follows [42]:

$$r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{(\sum_m \sum_n (A_{mn} - \bar{A})^2)(\sum_m \sum_n (B_{mn} - \bar{B})^2)}} , \quad (20)$$

for  $\bar{A}$  is the mean of the A matrix,  $\bar{B}$  is the mean of the B matrix, and m and n are the dimensions of matrices A and B, which are the two variables. In our case, these are the images themselves.  $\bar{A}$  and  $\bar{B}$  are therefore the mean of each element in the image.

Thresholding is performed on the final images to binarize them, simplifying the comparative analysis. Key characteristics analyzed are location of the centroid of the object, the ratio of the x and y bounds of the objects (the aspect ratio), and the difference of the areas between the simulated and captured images. These characteristics allow for a meaningful understanding of the differences between the object, both to the reader and purely quantitatively. Code used for this analysis is found in the Appendix. Thresholded versions of the examples provided (as captured by the point-electrode tank) are shown in Figure 5.3.

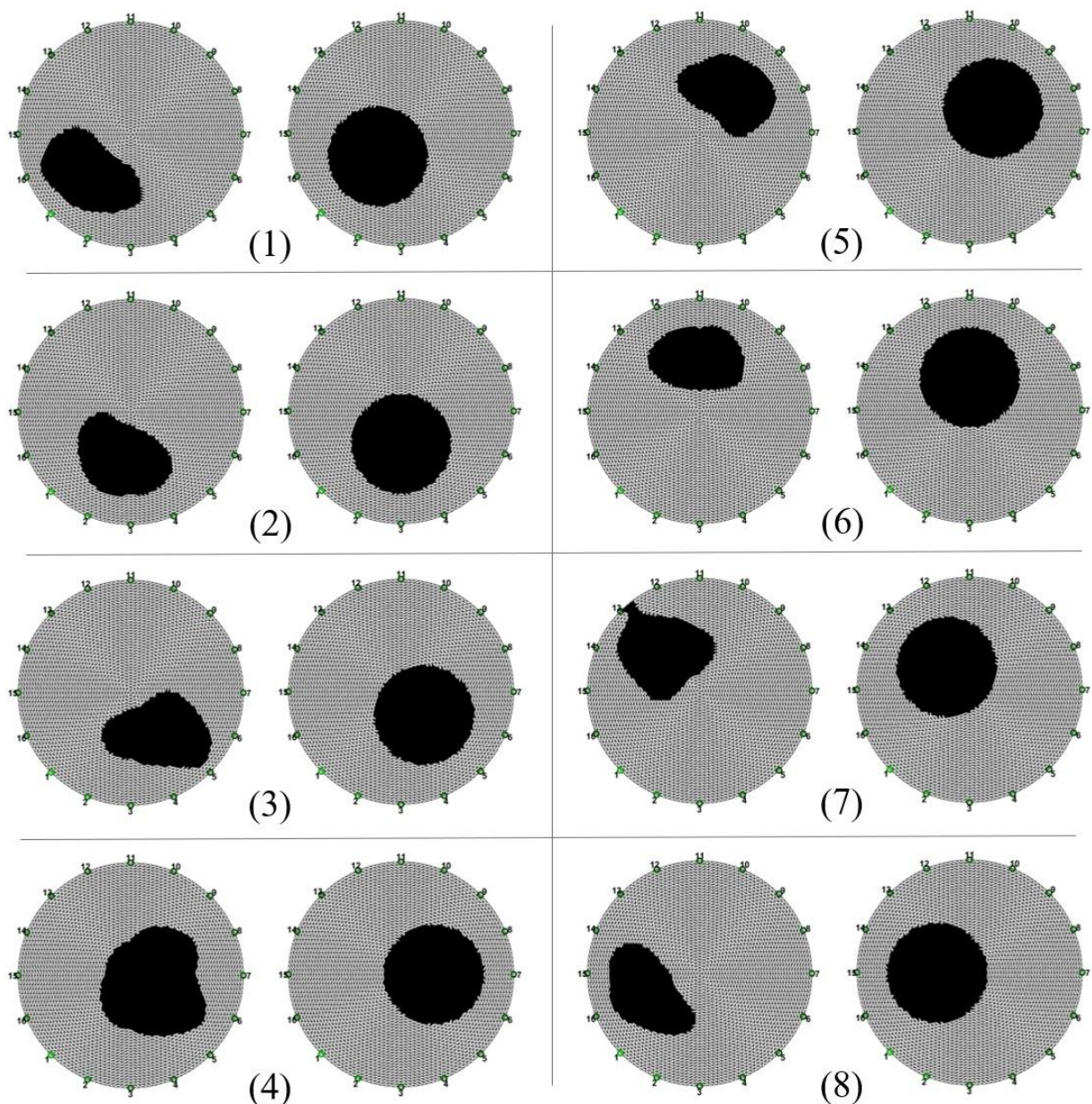


Figure 5.3. Point-electrode tank object reconstruction after thresholding, in 8 example positions (left) with the equivalent simulated object (right). Thresholding was set to -50 for the range shown in Figure 5.2. All reconstructed images produced in this thesis are made using injection signals of 0.1V AC + 0.1V DC, at 10kHz, as produced by the RP. Offsetting is required to maintain all signals greater than 0V to remain in the multiplexer operation voltage range of 0 to 5V. Output current amplitude produced from the VCCS using this RP output signal is 25.7 $\mu$ A.

The plate-electrode tank was not used in this analysis, as all resulting images showed poor correlation and were not deemed clinically viable. Comparison of electrode types remains for future work.

## 6. Results

All results are given as mean value and standard deviation of the samples.

The mean SNR of the system using the point-electrode tank before filtering was 31.616 dB (11.347), and 34.176 dB (11.98).

The final images shown in Figures 5.2 (and equivalent plate-electrode tank data) were binarized and shown in Figures 5.3 and 5.4, respectively. 2D Pearson coefficient between the simulated and final images was 0.702 (0.055) for the point-electrode tank, while the plate-electrode tank showed inconclusive images and non-correlative data. Using the point-electrode tank, the change in distance between centroid locations was 10.09mm (2.32). Percent difference of area and the aspect ratio were 22.1% (12.79) and 1.359 (0.165), respectively. To provide a rough metric of the plate-electrode tank performance, the difference in area between the reconstructed and simulated objects was nearly 98%.

Final results are listed together in Table 2.

Table 2. Data Construction Quality Characteristics

Tank	r	SNR (dB)		Reconstructed Object Results		
		Before Filtering	After Filtering	% Difference in Area	Centroid Distance (mm)	Aspect Ratio
Point Electrode	0.702 (0.055)	31.616 (11.347)	34.176 (11.98)	22.21 (12.79)	10.09 (2.32)	1.359 (0.165)

The plate-electrode tank showed generally negative results, and was not included in numerical analysis. An example image comparison as with the point-electrode tank is shown in Figure 5.5.

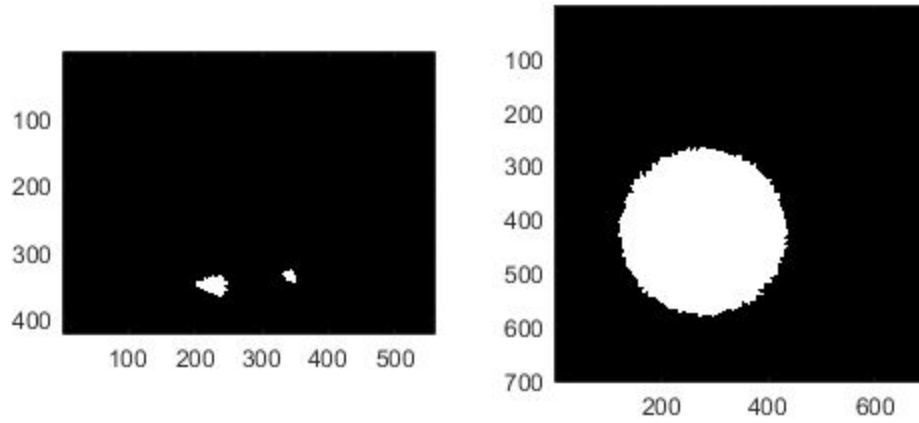


Figure 5.4. Plate-electrode tank thresholding results. Over 98% of the reconstructed area (left) to simulated area (right) is mismatched, and is unusable for imaging. Causes of error appear to be plate oxidation and poor signal transmission, likely on the electrode connection.

Total Variation Regularization was initially considered as a viable method to remove noise in the data prior to reconstruction, however results were not conclusive, and appeared to dramatically shift the resulting image. See Figure 5.5.

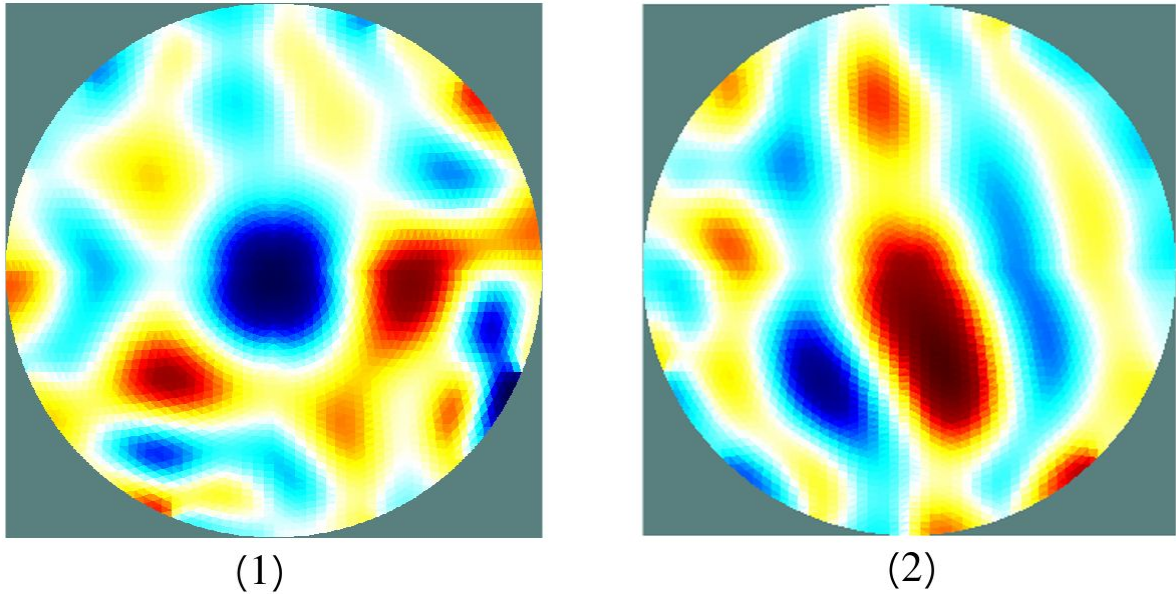


Figure 5.5. TVR Filtering (1) An example object reconstruction without using TVR. (2) The same data, filter with TVR, and reconstructed similarly. Note that 2 objects should be visible on the right image as with the left image.

This suggests that although TVR may not be suitable for pre-processing data prior to reconstruction, it may still be useful to incorporate into post-processing the image data. TVR is typically most suitable for impulse noise, which suggests that if it is incorporated somewhere in the reconstruction pipeline, where most of the impulse noise can be found, there may be a suitable step in which to raise SNR by filtering the data with TVR [15].

## 7. Evaluation

### 7.1. System Advantages

Our system has shown a relatively high potential SNR, compared to the commercial systems previously. Using the point-electrode tank, we achieved a SNR of 34.176 (11.98), compared to the Sheffield Mk 3.5 similar 40dB. However, some systems were able to achieve even higher SNR values, such as the Goe-MF II system's SNR of around 80dB [12], [43]. These systems are expensive and not battery-powered, so some losses in SNR should be expected of a portable system. Many factors can be independently addressed, including the equipment, protocols and reconstruction algorithms, and the currents used for injection, and the ability to interchange these parts is a benefit of our system.

Accuracy is also relatively high compared to the work of Xu [12], which used an 8-channel EIT system that was also based on the RP. Our point-electrode tank gave a 2D Peterson coefficient of 0.702, which was higher than Xu's value range of 0.675 to 0.69, while the plate-electrode tank showed virtually no correlation. In the instance of higher SNR values, we have shown that although accuracy is still limited when compared to systems such as CT and MRI, there is still value in identifying large objects and subvolumes in the body, such as lung cavities and gastrointestinal pockets, which may be the preferred method given the ease of use and minimal intrusiveness of EIT over other modalities.

In addition to SNR and accuracy, our system has shown a number of advantages over most other systems. First, our system is highly portable. Although a wall-mounted power supply was used to help eliminate any background noise during testing, the power supply could easily be replaced with a battery-based power source, removing even more 60Hz noise from the electrical lines supplying power.

Our system has been limited in its speed of acquiring data so as to minimize noise from the circuitry and environment, however in practical applications where this may not apply to this extent, the speed may be increased. The current speed (16 minutes to acquire both an empty tank and test configuration) is still roughly equivalent to the time of a CT scan when including preparation such as contrast administration, and is far less than the time needed for MRI acquisition. This shows the importance of consideration for EIT in future imaging, as the time to acquire images may be quite rapid.

Tissue type recognition and the state of tissue should also be considered as advantages to EIT imaging. The ability to identify inflamed tissue and tissue in which excess water has been collected could be of considerable aid to the quick and easy identification of conditions such as pneumonia, as well as other diseases with which pneumonia is associated.

Channel number is relatively high for a portable system, and is double the channels used by the previous work by Xu. This provided higher resolution, without obvious detriment to the system.

Tank design is simple to redesign, given the modular nature of our system. In our case, two different tanks were designed from scratch, and allowed us to see how different electrodes affected each stage of the process to image reconstruction. Additional tank designs could be incorporated into the design, as could electrodes. This includes the ability to use non-tank electrode sets as well, such as an

abdominal girdle, or smaller band, in which the electrodes are embedded, and through which a direct interface could be made with various parts of a patient's body.

Our system separates out the equipment used for data collection and image reconstruction, enabling a portable system to technically not require high-grade computing equipment regardless of future advancements, as reconstruction can take place remotely, even at a later time or date. However, at this stage, a laptop PC is generally more than capable of processing and reconstructing the data.

## **7.2. Limitations**

One of the primary limitations of our system, and a natural limitation of EIT in general, is the low resolution in comparison to other modalities. As channels are increased in the system design, however, and as reconstruction algorithms are improved and better implemented, the limitation may be mitigated.

The relatively high number of channels is also expandable in the system's current state. Without a complete overhaul or redesign, one way to continue the work of this research simply would be to incorporate the other RF channel pair of the RP (which remained unused for our project). Given that 1 RP channel was able to be successfully multiplexed out to 16 channels functionally, it would follow that to do the same with the second channel could be an easy way to double the number of functional channels. Although this would increase acquisition time, it would also increase the resolution and accuracy of the images.

The current used for injection can also be increased. Our currents were limited to a RMS value of 18uA. As mentioned previously, this is below the 100uA limit for bodily tissues. Improving the injection current would boost the signal, which in turn would likely raise SNR.

System voltage could be considered a limitation, as the absolute value of the system voltage was limited to 5 volts. A higher voltage source could still be provided via a battery, maintaining the system's portability, and potentially increasing the ability to inject higher values of current. This is left to future research.

Better characteristics of the custom circuitry built for this project would also be desirable, particularly the amplifying and filtering circuitry used to electrically post-process the signals from the EIT tank. However, amplifier and filter circuitry did not appear to be a major source of noise.

Despite the relatively quick acquisition time, this could be improved as well. Each signal injected is only a single 10kHz pulse, meaning that hypothetically, a full acquisition of the tank could be performed in  $16 * 16 * (1/10\text{kHz}) = 25.6\text{ms}$ . In our system's current state, time is allowed for conditions in the tank to stabilize, and maximize the SNR of the final result. As SNR improves in EIT, this time may be cut down significantly, further speeding up EIT acquisition times.

Differential imaging is one of the principles used by our system. This means that although we can image relatively quickly and easily, we still need the data from an empty tank to compare to. In future systems, this may be side-stepped by using an internal calibration system, or known object sets, to provide the "empty" tank data artificially. In some instances, differential imaging may be perfectly acceptable. In lung imaging, for instance, function can be measured as the differences between the observed data sets, regardless of what state each data set may contain on its own.

Manual intervention is still inherent to the image reconstruction portion. This is not technically a limitation of the data acquisition design, but rather of the reconstruction process. The aforementioned "k-value" needs to be determined manually for best results, and to optimize this is beyond the scope of this work. Recommendations would include mapping of optimal k-values in the future, especially with regard to specific configurations used to acquire data such as torso, abdomen, arm, etc.

Lastly, the method of injection and measurement is most likely not optimized. Although we measure adjacent channel pairs sequentially, some sources have claimed that better methods exist [44]. For instance, injection between adjacent pairs could be maintained, while the iterations increase such that injections are made by pairs across and slightly offset from one another, similar to a star pattern being drawn. Some works have cited the "next-to-opposite" methods to increase the image quality [45].

## **8. Future Work**

Many attempts to improve quality resulted in unstable results, but also presented distinct problems to focus on in the future. In addition to recommendations noted in the Evaluation section, a potential approach using specialized electrodes is provided as follows.



### **8.1. Electrodes**

Part of the reason that the plate electrodes did not result in improvements to image quality was oxide formation due to the copper plating used. Improvements could be made by using non-oxide forming materials, or materials which have electrically-conducting oxides like silver. Alternatively, it may be of greater benefit to be able to process each electrode individually, effectively granting the ability to recognize and correct for different values of conductivity and process each signal independently.

An example of such a system could be implemented by using inexpensive microcontrollers to output digital signals and observe the response in real-time through each electrode directly. Information could then be accumulated and mixed together digitally.

### **8.2. Digital Multiplexing**

If the electrodes were replaced as purely physical entities with complete processors to collect and mix the data, the data would be able to be mixed digitally. Some microcontrollers are able to simultaneously interface with upwards of 32 input/output channels, meaning that for less than or equal to a 32-channel EIT system, all channels could be collected and multiplexed digitally, in real-time, and with the ability to apply digital filters on the data prior to mixing. This could be useful in correcting for differences in individual electrodes, and for reducing noise prior to reconstructing the data.

### **8.3. Real-Time Microcontroller Processing**

Continuing the idea of self-processing electrodes, all incoming data would be able to be collected together in parallel by a sufficiently capable microcontroller device and then piped into a computer which could process the data, performing all advanced filtering and reconstruction on a PC, while the data would still continue to be collected via the real-time microcontroller system.

Reconstructed and raw data could both be gathered continuously, for however long a scan is needed. The data could be transferred in full, to whichever platforms are to perform reconstruction or viewing.

## **9. Conclusions**

EIT is still an emerging technology, and many issues remain to be addressed.

Reconstruction methodologies are constantly being improved, and advancements in neural networks make for a logical next step to create large-scale improvements in image reconstruction. As the number of usable electrodes, processing speeds, and system efficiency and SNR improve, EIT will likely begin to find its way into research and clinical applications alongside current and future imaging modalities.

## APPENDIX

### MATLAB Code

Code written specifically for this project is described in this Appendix.

The script for testing the RP through the two-electrode mini-tank, "RP\_2\_electrode\_tank\_test.m," is as follows:

#### RP\_2\_electrode\_tank\_test.m

```
clc
clear all
close all

%% Set connection with Red Pitaya
IP = '132.239.135.14';
port = 5000;
tcpipObj=tcpip(IP, port);
% Set buffer size. % A read operation is terminated if the amount of data
% stored in the input buffer equals the InputBufferSize value.
tcpipObj.InputBufferSize = 16384*32;
tcpipObj.OutputBufferSize = 16384*32;

%% Open connection with Red Pitaya
fopen(tcpipObj);
% Terminator is carriage return followed by line feed
tcpipObj.Terminator = 'CR/LF';
% Clear data from input and output buffer:
flushinput(tcpipObj)
flushoutput(tcpipObj)

%% Set signal generation parameters
fprintf(tcpipObj,'GEN:RST');
% Use a 0.5V, 5Hz sinusoid:
fprintf(tcpipObj,'SOUR2:FUNC SINE');
fprintf(tcpipObj,['SOUR2:FREQ:FIX ',num2str(5)]);
fprintf(tcpipObj,['SOUR2:VOLT ',num2str(0.5)]);

fprintf(tcpipObj,'SOUR2:BURS:STAT ON'); % Set burst mode to ON
fprintf(tcpipObj,'SOUR2:BURS:NCYC 5'); % Set 1 cycle of sine wave
fprintf(tcpipObj,'SOUR2:BURS:NOR 1'); % Set 1 pulse of sine wave

%% Set acquisition parameters
fprintf(tcpipObj,'ACQ:RST');
% Decimation factor:
```

```

fprintf(tcpipObj,'ACQ:DEC 8192');
% Set trigger level and delay (in samples).
% A trigger delay of 0 places the start of
% the signal in the center of the frame,
% while the signal starts at the left of the
% frame for a delay of 8192 samples.
fprintf(tcpipObj,'ACQ:TRIG:LEV 0');
dly = 7500;
fprintf(tcpipObj,['ACQ:TRIG:DLY ',num2str(dly)]);

%% Start generating signal
fprintf(tcpipObj,'OUTPUT2:STATE ON');

%% Start acquiring data
fprintf(tcpipObj,'ACQ:START');
pause(1); % Wait for buffer to fill with data
% Set generator trigger to immediately:
fprintf(tcpipObj,'ACQ:TRIG AWG_PE');
fprintf(tcpipObj,'SOUR2:TRIG:IMM');

%% Wait for trigger
while 1
    trig_rsp=query(tcpipObj,'ACQ:TRIG:STAT?');
    if strcmp('TD',trig_rsp(1:2))
        break
    end
end

%% Read & plot
signal_str=query(tcpipObj,'ACQ:SOUR2:DATA?');
% Remove first character in string "{"
% and the last two characters, corresponding to
% empty space and "}".
signal_num=str2num(signal_str(1,2:length(signal_str)-3));

%% Stop generating signal
fprintf(tcpipObj,'OUTPUT2:STATE OFF');

%% Plot the signal, with cursors indicating the
%% signal maximum and period
figure
plot_prefilt = plot(signal_num);
hold on
title('Recorded signal');
grid minor
[maxval,imax] = max(signal_num);
[minval,imin] = min(signal_num);
xlabel('Data point #');

```

```

ylabel('Amplitude (V)');
x=1:16384;
maxval(1:16384)=maxval;
minval(1:16384)=minval;
plot(x,maxval,'--k')
plot(x,minval,'--k','Linewidth',0.8)
txt1 = ['Max = ',num2str(maxval(1)), 'V'];
txt2 = ['Min = ',num2str(minval(1)), 'V'];
text(1000,maxval(1)+0.04,txt1)
text(1000,minval(1)-0.04,txt2)
ylim([-0.65,0.65]);
line([imax imax],[-1 1])
fs = 15.2e3; % Sampling frequency
numsamp = fs/f;
line([imax-numsamp imax-numsamp],[-1 1])
txt3 = ['N1 = ',num2str(imax-numsamp), '\rightarrow'];
txt4 = ['\leftarrow N2 = ',num2str(imax)];
text(imax-numsamp-3000,maxval(1)+0.04,txt3)
text(imax+100,maxval(1)+0.04,txt4)

%% Close connection with Red Pitaya
fclose(tcpipObj);

```

The script “hardware\_control.m” is written to perform all functions needed to capture data from the EIT hardware and save it for reconstruction.

### **hardware\_control.m**

```

%% HARDWARE CONTROL CODE
% This Matlab code is used for signal generation and acquisition with the
% RP STEMLab
clc; clear all; close all

%% Path for saving data
% 68 is starting (null set)
testNumber = 80;
path = ['C:\Users\Andrew\Documents\MATLAB'];
path = ['C:\Users\Andrew\Documents\MATLAB\EIT_project\Andrew\test',num2str(testNumber)];
if ~exist(path, 'dir') % Create directory if it doesn't exist
    mkdir(path)
else % Cancel or overwrite directory if it already exists
    resp = input('Directory already exists - do you wish to continue? [Y/N] >>','s');
    if resp == 'N' || resp == 'n'
        error('Session aborted')
    else
        mkdir(path)
    end
end

```

```

    end
end

% Create a README file with the session parameters
fid = fopen([path,'README.txt'],'wt');
fprintf(fid,'%s\n',date);
fprintf(fid,'Test Number %d:\n\n',testNumber);

%% Set connection with Red Pitaya
%IP = '132.239.135.14'; % Input IP of the Red Pitaya
IP = '192.168.128.1'; % Input IP of the Red Pitaya
port = 5000;
tcpipObj=tcpip(IP, port);
% Set buffer size. % A read operation is terminated if the amount of data
% stored in the input buffer equals the InputBufferSize value.
tcpipObj.InputBufferSize = 16384*32;
tcpipObj.OutputBufferSize = 16384*32;

%% Open connection with Red Pitaya
fopen(tcpipObj);
tcpipObj.Terminator = 'CR/LF'; % Terminator is carriage return followed by line feed
flushinput(tcpipObj) % Remove data from input buffer
flushoutput(tcpipObj) % Remove data from output buffer

%% Set digital pins to output
set_pins_output(tcpipObj); % Digital pins for MUX addressing

%% Set signal generation parameters
freq = 10e3; % Frequency in Hz
amp = 0.1; % Amplitude in V
ost = 0.1; % Offset in V
% Generate signal:
set_gen(tcpipObj,freq,amp,ost,-1,1); % Parameters: RP_object,freq,ampl,ncyc,nbursts
% --> Generates an infinite number of 0.1V amplitude,
% 0.1V offset 10kHz bursts

%% Set signal acquisition
DEC = 8; % Decimation factor
trig_dly = 8192; % Trigger delay trig_dly = 8192 puts start of signal on leftmost side
set_acq(tcpipObj,DEC,0,trig_dly); % Parameters: RP_object,dec,trig_lvl,trig_dly

%% Channel injection through each electrode pair
rms_signal = []; % rms vector of each measurement signal
rms_signal_filt = []; % rms vector of filtered measurements

% Loop through each electrode for injection
for i=0:15
    i_MUX1 = i; % Y_i Channel for MUX1

```

```

% MUX2 channel should be adjacent to MUX1 channel
if i_MUX1 == 15
i_MUX2 = 0;
else
i_MUX2 = i_MUX1+1;
end

% Get 4-bit binary representation of channel number addressed by MUX 1&2
i_binary_MUX1 = dec2bin(i_MUX1,4);
i_binary_MUX2 = dec2bin(i_MUX2,4);

% Select the given channels for injection
set_channel_MUX(tcpipObj,'MUX1',i_binary_MUX1); % Parameters: RP object, MUX label, MUX
channel select
set_channel_MUX(tcpipObj,'MUX2',i_binary_MUX2);

disp(['Injecting at channels ',num2str(i_MUX1),' and ',num2str(i_MUX2)]);

% Generate signal
fprintf(tcpipObj,'OUTPUT1:STATE ON'); % Set output to ON

%%%%% Plot signals if needed
% figure;
% sgtitle(['Input electrode #',num2str(i+1)]);
%%%%%

%% Measure voltage across each electrode pair
for j=0:15
i_MUX3 = j; % Choose measuring channel for MUX3

% MUX4 channel should be adjacent to MUX3 channel
if i_MUX3 == 15
i_MUX4 = 0;
else
i_MUX4 = i_MUX3+1;
end

% 4-bit binary representation of channel number addressed by MUX 3&4
i_binary_MUX3 = dec2bin(i_MUX3,4);
i_binary_MUX4 = dec2bin(i_MUX4,4);

% Select the given channels to measure
set_channel_MUX(tcpipObj,'MUX3',i_binary_MUX3);
set_channel_MUX(tcpipObj,'MUX4',i_binary_MUX4);

disp(['Measuring at channels ',num2str(i_MUX3),' and ',num2str(i_MUX4)]);

```

```

%% Acquire signal
% Start acquiring signal
pause_time = 1; % Pause to fill buffer
signal_num = start_acq(tcpipObj,pause_time); % Parameters: RP object, pause time

% Save signal
signal_allchannels(i+1,j+1,:) = signal_num; % Measurement across all electrode pairs for all input
pairs
                                % electrode_index = channel_index+1

%% Process signal
signal_num_filt = signal_num(:)-mean(signal_num(:)); % Remove mean from signal
% Low-pass filter the signal - Parameters: signal, cut-off frequency, sampling frequency,
steepness of filter
signal_num_filt = lowpass(signal_num_filt,11e3,15.6e6,'Steepness',0.95);
signal_allchannels_filt(i+1,j+1,:) = signal_num_filt; % Measurement across all electrode pairs for
all input pairs

%%%%%%%% Plot signal if needed
% subplot(4,4,i_MUX3+1);
% plot(signal_num_filt);
% title(['Signal at electrode #',num2str(i_MUX3+1)]);
%%%%%%%%

% Calculate rms value of signal
rms_signal = [rms_signal,rms(signal_num-mean(signal_num))]; % Remove mean from signal and
calculate rms value
rms_signal_filt = [rms_signal_filt,rms(signal_num_filt)]; % Calculate rms value of signal
pause(0.2); % Allows signals to settle (Montellano et al., 2011)
end
fprintf(tcpipObj,'OUTPUT1:STATE OFF'); % Set output to OFF at end of injection
pause(0.5); % Allows signals to settle (Montellano et al., 2011)
end

%% Save signals
save([path,'signal_allchannels.mat'],'signal_allchannels');
save([path,'signal_allchannels_filt.mat'],'signal_allchannels_filt');
save([path,'rms_signal.mat'],'rms_signal');
save([path,'rms_signal_filt.mat'],'rms_signal_filt');

%% Save signal parameters in README file
fprintf(fid, 'Input signal parameters:\n');
fprintf(fid, 'Frequency: %d Hz\n', freq);
fprintf(fid, 'Amplitude: %d V\n', amp);
fprintf(fid, 'Offset: %d V\n', ost);

%% Close connection with Red Pitaya
fclose(tcpipObj);

```



```
fclose(fid);
```

Additionally, functions used to support “hardware\_control.m” include the following:

### **set\_acq.m**

```
function [outputArg1,outputArg2] = set_acq(tcpipObj,dec,trig_lvl,trig_dly)
%SET_ACQ Summary of this function goes here
% Detailed explanation goes here

fprintf(tcpipObj,'ACQ:RST');
fprintf(tcpipObj,['ACQ:DEC ',num2str(dec)]);
fprintf(tcpipObj,['ACQ:TRIG:LEV ',num2str(trig_lvl)]);
fprintf(tcpipObj,['ACQ:TRIG:DLY ',num2str(trig_dly)]);

end
```

### **set\_channel\_MUX.m**

```
function set_channel_MUX(tcpipObj,MUXi,i_binary_MUXi)
%SET_CHANNEL_MUX Summary of this function goes here
% Detailed explanation goes here

if MUXi=='MUX1'
    fprintf(tcpipObj,['DIG:PIN DIO0_N,' num2str(i_binary_MUXi(4))]); % LSB
    fprintf(tcpipObj,['DIG:PIN DIO0_P,' num2str(i_binary_MUXi(3))]);
    fprintf(tcpipObj,['DIG:PIN DIO1_N,' num2str(i_binary_MUXi(2))]);
    fprintf(tcpipObj,['DIG:PIN DIO1_P,' num2str(i_binary_MUXi(1))]); % MSB

elseif MUXi=='MUX2'
    fprintf(tcpipObj,['DIG:PIN DIO2_N,' num2str(i_binary_MUXi(4))]); % LSB
    fprintf(tcpipObj,['DIG:PIN DIO2_P,' num2str(i_binary_MUXi(3))]);
    fprintf(tcpipObj,['DIG:PIN DIO3_N,' num2str(i_binary_MUXi(2))]);
    fprintf(tcpipObj,['DIG:PIN DIO3_P,' num2str(i_binary_MUXi(1))]); % MSB

elseif MUXi=='MUX3'
    fprintf(tcpipObj,['DIG:PIN DIO4_N,' num2str(i_binary_MUXi(4))]); % LSB
    fprintf(tcpipObj,['DIG:PIN DIO4_P,' num2str(i_binary_MUXi(3))]);
    fprintf(tcpipObj,['DIG:PIN DIO5_N,' num2str(i_binary_MUXi(2))]);
    fprintf(tcpipObj,['DIG:PIN DIO5_P,' num2str(i_binary_MUXi(1))]); % MSB

elseif MUXi=='MUX4'
    fprintf(tcpipObj,['DIG:PIN DIO6_N,' num2str(i_binary_MUXi(4))]); % LSB
    fprintf(tcpipObj,['DIG:PIN DIO6_P,' num2str(i_binary_MUXi(3))]);
    fprintf(tcpipObj,['DIG:PIN DIO7_N,' num2str(i_binary_MUXi(2))]);
    fprintf(tcpipObj,['DIG:PIN DIO7_P,' num2str(i_binary_MUXi(1))]); % MSB

end
```

end

### set\_gen.m

```
function set_gen(tcpipObj,freq,ampl,ost,ncyc,nbursts)
%GEN_SIN Summary of this function goes here
% Detailed explanation goes here
    fprintf(tcpipObj,'GEN:RST');           % Resets generator settings to default
    fprintf(tcpipObj,'SOUR1:FUNC SINE');   % Set waveform of fast analog outputs

    fprintf(tcpipObj,['SOUR1:FREQ:FIX ',num2str(freq)]); % Set frequency of output signal
    fprintf(tcpipObj,['SOUR1:VOLT ',num2str(ampl)]);     % Set amplitude of output signal
    fprintf(tcpipObj,['SOUR1:VOLT:OFFS ',num2str(ost)]);

    fprintf(tcpipObj,'SOUR1:BURS:STAT ON'); % Set burst mode to ON
    fprintf(tcpipObj,['SOUR1:BURS:NCYC ',upper(num2str(ncyc))]); % Set NCYC pulse of sine wave
    fprintf(tcpipObj,['SOUR1:BURS:NOR ',upper(num2str(nbursts))]); % Set number of repeated
bursts
end
```

### set\_pins\_output.m

```
function set_pins_output(tcpipObj)
% set_pins_output takes as an input the RP object and sets all digital pins
% to output.

% Pins setting channel for MUX 1
fprintf(tcpipObj,'DIG:PIN:DIR OUT,DIO0_N'); % Pin 00
fprintf(tcpipObj,'DIG:PIN:DIR OUT,DIO0_P'); % Pin 01
fprintf(tcpipObj,'DIG:PIN:DIR OUT,DIO1_N'); % Pin 10
fprintf(tcpipObj,'DIG:PIN:DIR OUT,DIO1_P'); % Pin 11

% Pins setting channel for MUX 2
fprintf(tcpipObj,'DIG:PIN:DIR OUT,DIO2_N'); % Pin 20
fprintf(tcpipObj,'DIG:PIN:DIR OUT,DIO2_P'); % Pin 21
fprintf(tcpipObj,'DIG:PIN:DIR OUT,DIO3_N'); % Pin 30
fprintf(tcpipObj,'DIG:PIN:DIR OUT,DIO3_P'); % Pin 31

% Pins setting channel for MUX 3
fprintf(tcpipObj,'DIG:PIN:DIR OUT,DIO4_N'); % Pin 40
fprintf(tcpipObj,'DIG:PIN:DIR OUT,DIO4_P'); % Pin 41
fprintf(tcpipObj,'DIG:PIN:DIR OUT,DIO5_N'); % Pin 50
fprintf(tcpipObj,'DIG:PIN:DIR OUT,DIO5_P'); % Pin 51

% Pins setting channel for MUX 4
fprintf(tcpipObj,'DIG:PIN:DIR OUT,DIO6_N'); % Pin 60
```

```
fprintf(tcpipObj,'DIG:PIN:DIR OUT,DIO6_P'); % Pin 61
fprintf(tcpipObj,'DIG:PIN:DIR OUT,DIO7_N'); % Pin 70
fprintf(tcpipObj,'DIG:PIN:DIR OUT,DIO7_P'); % Pin 71
```

```
end
```

### **start\_acq.m**

```
function signal_num = start_acq(tcpipObj,pause_time)
%START_ACQ Summary of this function goes here
% Detailed explanation goes here
fprintf(tcpipObj,'ACQ:START');
pause(pause_time);
fprintf(tcpipObj,'ACQ:TRIG AWG_PE');
fprintf(tcpipObj,'SOUR1:TRIG:IMM');          % Set generator trigger to immediately

%% Wait for trigger
while 1
    trig_rsp=query(tcpipObj,'ACQ:TRIG:STAT?');
    if strcmp('TD',trig_rsp(1:2)) % Wait until trigger (then response is 'TD', else it is 'wait')
        break
    end
end

% Read
signal_str=query(tcpipObj,'ACQ:SOUR1:DATA?'); % Read full buffer size starting from trigger
signal_num=str2num(signal_str(1,2:length(signal_str)-3)); % First character in string is {
                                                    % and 2 latest are empty spaces and last is }.

end
```

Coding for image reconstruction using the data acquired by “hardware\_control.m” is found in “image\_reconstruction.m.”

### **image\_reconstruction.m**

```
% Image reconstruction
clear all; close all; clc;
run 'C:\Users\Andrew\Documents\MATLAB\EIT_project\eidors-v3.9.1\eidors\startup.m'

%% Load data
% Load reference data (homogeneous)
path_ref = 'C:\Users\Andrew\Documents\MATLAB\EIT_project\Andrew\test60';
addpath(path_ref)
load rms_signal_filt
%load rms_signal_filt_TVR_100x_a
v_ref= real(rms_signal_filt);
%v_ref= real(u);
clear rms_signal_filt
```

```

%clear u
%clear rms_signal_filt_TVR_100x_a

% Load object data (inhomogeneous)
path_obj = 'C:\Users\Andrew\Documents\MATLAB\EIT_project\Andrew\test64';
%path_obj = 'C:\Users\Andrew\Documents\MATLAB\EIT_project\Andrew\2020-01-27-null-tank\test50';
addpath(path_obj)
load rms_signal_filt
%load rms_signal_filt_TVR_100x_b
v_obj= real(rms_signal_filt);
%v_obj= real(u);
clear rms_signal_filt
%clear u
%clear rms_signal_filt_TVR_100x

%% Plot EIT measurements
plot(v_ref)
hold on;
plot(v_obj)
hold off;
title('EIT measurements')
legend('Without object','With object');
xlabel('Measurement index');
ylabel('RMS Voltage');
xlim([0,256])

%% Get a 2D image reconstruction model
% Generate the forward model fwd model, which is designed to represent the finite element model (FEM),
% electrode positions and properties, and stimulation patterns
imdl = mk_common_model('j2c'); % 2D circ model (2304 elems) - lowercase 'c' for point electrodes
[stim,mse1]=mk_stim_patterns(16,1,[1 0],[1 0],{'no_meas_current'},0.3); % Select stimulation patterns
imdl.fwd_model.stimulation = stim;
imdl.fwd_model.meas_select = mse1;
imdl.reconst_type = 'difference'; % Differential imaging
% Change location of electrode numbering to match our experimental setup
imdl.fwd_model.electrode([7:16,1:6])= imdl.fwd_model.electrode; % For electrode 1 on bottom-left
imdl.fwd_model.electrode([1,16:-1:2]) = imdl.fwd_model.electrode; % For counter-clockwise

% Show forward model
fem_model = mk_image(imdl);
figure;
options(1) = 0;
options(2) = 1;
show_fem(fem_model,options);
set(gca,'xtick',[])
set(gca,'xticklabel',[])
set(gca,'ytick',[])
set(gca,'yticklabel',[])

```

```

title({'FE model for the circular tank',' '})
axis off

% Removing faulty electrodes (optional):
% imdl.meas_icov = meas_icov_rm_elects(imdl, [7,8]);

%% Reconstruct images
figure
for idx= 1:3
    if idx==1
        imdl.hyperparameter.value= 0.01; % Hyperparameter required for regularization
    elseif idx==2
        imdl.hyperparameter.value= .05;
    elseif idx==3
        imdl.hyperparameter.value= 1;
    end
    % Solve inverse problem
    img = inv_solve(imdl, v_ref, v_obj); % Default: 'inv_solve_diff_GN_one_step'

    % Show reconstructed image
    subplot(1,3,idx);
    % show_slices(img);
    show_fem(img,options);
    title(['Reconstructed image for k = ',num2str(imdl.hyperparameter.value)],' ');
    set(gca,'xtick',[])
    set(gca,'xticklabel',[])
    set(gca,'ytick',[])
    set(gca,'yticklabel',[])
    axis off
    set(gca,'FontSize',14)
end

% Save image for k=0.01
%imdl.hyperparameter.value= 0.01;
% Save image for k= custom
imdl.hyperparameter.value= 0.001;

img = inv_solve(imdl, v_ref, v_obj);
img.calc_colours.npoints = 701; % pixel width/height to map to - odd number for center point

figure;
Reconstructed_image_FEM = show_fem(img,options);
set(gca,'xtick',[])
set(gca,'xticklabel',[])
set(gca,'ytick',[])
set(gca,'yticklabel',[])
axis off
saveas(gcf,[path_obj,'Reconstructed_image_FEM.png'])

```

```
figure;
Reconstructed_image = show_slices(img);
cmap = colormap;

% imwrite(Reconstructed_image_FEM,cmap,[path_obj,'Reconstructed_image_FEM.png']);
imwrite(Reconstructed_image,cmap,[path_obj,'Reconstructed_image.png']);
```

To obtain simulated images for accuracy assessment, we used the script “simulated\_image.m.”

### **simulated\_image.m**

```
clear all; close all; clc;
run 'C:\Users\Andrew\Documents\MATLAB\EIT_project\eidors-v3.9.1\eidors\startup.m'
path_save = 'C:\Users\Andrew\Documents\MATLAB\EIT_project\Andrew';

% 2D Model
imdl= mk_common_model('j2c');
imdl.fwd_model.electrode([7:16,1:6])= imdl.fwd_model.electrode; % For electrode 1 on bottom-left
imdl.fwd_model.electrode([1,16:-1:2]) = imdl.fwd_model.electrode; % For counter-clockwise

% Create a homogeneous image
img_1 = mk_image(imdl);
options(1) = 0;
options(2) = 1; % Show electrode numbers
show_fem(img_1,options);

img_1.elem_data(:) = 129; % Homogeneous conductivity

% Add a circular object
% Calculate element membership in object
d_obj = 5; % Object diameter in cm
d_tank = 11.3; % Tank diameter in cm
d_obj_n = d_obj/(d_tank/2); % Normalized object diameter - radius of tank in image is 1
r_obj = 1-d_obj_n/2-1.5/(d_tank/2); % Distance of object center from center of tank

% Location of object centroid
electrode_i = 1;
theta = 5*pi/4+2*pi*(electrode_i-1)/16; % Angle at which the object is located
x_obj = r_obj*cos(theta);
y_obj = r_obj*sin(theta);

% Inhomogeneous object
img_2 = img_1;
select_fcn = inline(['(x-',num2str(x_obj),').^2+(y-',num2str(y_obj),').^2<',num2str(d_obj_n/2),'^2'],'x','y','z');
% Find exact diameter and position
img_2.elem_data = 1 - elem_select(img_2.fwd_model, select_fcn); % Object has lower conductivity
```

```
img_2.calc_colours.npoints = 701; % pixel width/height to map to
```

```
figure;  
show_fem(img_2,options);  
set(gca,'xtick',[])  
set(gca,'xticklabel',[])  
set(gca,'ytick',[])  
set(gca,'yticklabel',[])  
axis off  
saveas(gcf,[path_save,'\Simulated_image_FEM.png'])
```

```
% Make image to print, without FEM mesh
```

```
figure;  
t = show_slices(img_2);  
cmap = colormap;
```

```
imwrite(t,cmap,[path_save,'\Simulated_image.png']);
```

Finally, in order to compare the simulated to the reconstructed images, we apply the script “image\_evaluation.m” as follows:

#### **image\_evaluation.m**

```
%%%%%%%%%%%%%  
%%%%%%%%%%%%  
%           Image segmentation code           %  
%%%%%%%%%%%%  
% Code description:  
% Identify object based on conductivity (color). Then, compare to simulated  
% object.  
  
close all; clear all; clc;  
  
%% Import reconstructed image  
path = 'C:\Users\Andrew\Documents\MATLAB\EIT_project\Andrew';  
  
[I,cmap] = imread([path,'\Reconstructed_image.png']);  
I_rgb = ind2rgb(I,cmap); % Convert image to RGB (initially colormap)  
  
% Extract image dimensions  
info = imfinfo([path,'\Reconstructed_image.png']);  
I_W = info.Width;  
I_H = info.Height;  
tank_pixel_W = I_W-2; % Tank contours start 1 pixel from each edge  
tank_pixel_H = I_H-2;
```

```

% Find pixel size from image dimensions
d_tank = 113; % Tank diameter in mm
pixel_size = d_tank/tank_pixel_W; % pixel size in mm

%% Import simulated image
[I_comp,cmap] = imread([path,"Simulated_image.png"]);
I_comp_rgb = ind2rgb(I_comp,cmap);

%% Calculate cross-correlation coefficient between images
% Extract central circle from image - ignore contours
[nx,ny,d] = size(I) ;
[X,Y] = meshgrid(1:ny,1:nx) ;
px = nx/2;
py = ny/2;
th = linspace(0,2*pi) ;
r = floor(nx/2); % Radius of circle - better take less to avoid comparing black borders
xc = nx/2+r*cos(th) ;
yc = ny/2+r*sin(th) ;
% Keep only points lying inside circle
idx = inpolygon(X(:),Y(:),xc',yc) ;

R = corr2(I(idx),I_comp(idx)); % Comparing image without borders

%% Show RGB images
figure;
subplot(1,2,1);
image(I_rgb);
axis image % Keep aspect ratio

subplot(1,2,2);
image(I_comp_rgb);
axis image

%% Extract object from reconstructed image
rImage = I_rgb(:, :, 1); % Red channel
gImage = I_rgb(:, :, 2); % Green channel
bImage = I_rgb(:, :, 3); % Blue channel

figure;
subplot(2,2,1);
rHist = histogram(rImage); % Histogram of red values
subplot(2,2,2);
gHist = histogram(gImage); % Histogram of green values
subplot(2,2,3);
bHist = histogram(bImage); % Histogram of blue values

%% Threshold reconstructed image to extract object

```



```

binary_Image = (I_rgb(:,:,3)>50/255)&(I_rgb(:,:,2)<80/255)&(I_rgb(:,:,1)<50/255);
binary_Image_2 = I(:,:,)<60;

figure;
subplot(1,2,1)
imagesc(binary_Image);
subplot(1,2,2)
imagesc(binary_Image_2);
title('test');

figure;
subplot(1,2,1)
imagesc(binary_Image);
colormap(gray) % Use grayscale
subplot(1,2,2)
image(I_rgb);

%% Threshold simulated image to extract object
rImage_comp = I_comp_rgb(:, :, 1);
gImage_comp = I_comp_rgb(:, :, 2);
bImage_comp = I_comp_rgb(:, :, 3);

figure;
subplot(2,2,1);
rHist_comp = histogram(rImage_comp);
subplot(2,2,2);
gHist_comp = histogram(gImage_comp);
subplot(2,2,3);
bHist_comp = histogram(bImage_comp);

binary_Image_comp =
(I_comp_rgb(:,:,3)>50/255)&(I_comp_rgb(:,:,2)<80/255)&(I_comp_rgb(:,:,1)<50/255);
figure;
subplot(1,2,1)
imagesc(binary_Image_comp);
colormap(gray)
subplot(1,2,2)
image(I_comp_rgb);

%% Compare reconstructed and simulated objects
figure;
subplot(1,2,1)
imagesc(binary_Image);
axis image
colormap(gray)
subplot(1,2,2)
imagesc(binary_Image_comp);
axis image

```

```

% Extract features from reconstructed and simulated images
stats_rec = regionprops(binary_Image,'all');
stats_sim = regionprops(binary_Image_comp,'all');

if numel(stats_rec)>1 % If detected more than one object, perform opening (Morphological Operation)
%     disp('Too many objects - opening image');
%     disk_radius = (stats_sim.BoundingBox(3)-stats_sim.BoundingBox(1))/2;
%     strel_radius = disk_radius*2/3;
%     se = strel('disk',round(strel_radius));
%     binary_Image_open = imopen(binary_Image,se);
%     stats_rec = regionprops(binary_Image_open);
%     disp('Too many objects - keeping largest')

    % Only keep largest object:
    maxArea = 0;
    idx_max = 0;
    for i=1:numel(stats_rec)
        if stats_rec(i).Area>maxArea
            idx_max = i;
            maxArea = stats_rec(i).Area;
        end
    end

    stats_rec = stats_rec(idx_max);
end

% Area in mm^2:
area_rec = stats_rec.Area*(pixel_size^2);
area_sim = stats_sim.Area*(pixel_size^2);
area_diff = (area_sim-area_rec)/area_sim*100;

% Distance between centroids:
dist_x = (stats_rec.Centroid(1)-stats_sim.Centroid(1))*pixel_size; % Distance along x, in mm
dist_y = (stats_rec.Centroid(2)-stats_sim.Centroid(2))*pixel_size; % Distance along y, in mm
euclid_dist = sqrt(dist_x^2+dist_y^2);

% Circularity
circularity_rec = (4 * pi * stats_rec.Area)./(stats_rec.Perimeter .^ 2);
circularity_sim = (4 * pi * stats_sim.Area)./(stats_sim.Perimeter .^ 2);
circularity_ratio = circularity_rec./circularity_sim;
aspect_ratio = stats_rec.MajorAxisLength./stats_rec.MinorAxisLength;

%% Results:
disp(['Cross-correlation coefficient: ',num2str(R)]);
disp(['Centroid distance: ',num2str(euclid_dist),'mm']);
disp(['Percentage difference in area: ',num2str(area_diff),'%']);
disp(['Circularity ratio: ',num2str(circularity_ratio)]);

```

```

disp(['Aspect ratio: ',num2str(aspect_ratio)]);

% Write results to file:
fid = fopen([path,'\Results.txt'],'wt');

fprintf(fid,'Results:\n\n');
fprintf(fid,'Cross-correlation coefficient: %f\n',R);
fprintf(fid,'Object:\n');
fprintf(fid,'- Centroid distance: %f mm\n',euclid_dist);
fprintf(fid,'- Percentage difference in area: %f%%\n',area_diff);
fprintf(fid,'- Circularity ratio: %f\n',circularity_ratio);
fprintf(fid,'- Aspect ratio: %f\n\n',aspect_ratio);

% Save binary images
imwrite(~binary_Image,[path,'Reconstructed_image_object.png']);
imwrite(~binary_Image_comp,[path,'Simulated_image_object.png']);

fclose(fid);

```

## REFERENCES

- [1] Smith, N. B., & Webb, A. (2010). *Introduction to medical imaging: physics, engineering and clinical applications*. Cambridge university press.
- [2] Lin, E. C. (2010, December). Radiation risk from medical imaging. In *Mayo Clinic Proceedings* (Vol. 85, No. 12, pp. 1142-1146). Elsevier.
- [3] Hollingsworth, K. G. (2015). Reducing acquisition time in clinical MRI by data undersampling and compressed sensing reconstruction. *Physics in Medicine & Biology*, 60(21), R297.
- [4] Cline, H. E., & Anthony, T. R. (1994). *U.S. Patent No. 5,291,890*. Washington, DC: U.S. Patent and Trademark Office.
- [5] Adler, A., & Boyle, A. (2017). Electrical impedance tomography: Tissue properties to image measures. *IEEE Transactions on Biomedical Engineering*, 64(11), 2494-2504.
- [6] Schmölder, G. M., Bhatia, R., Davis, P. G., & Tingay, D. G. (2013). A comparison of different bedside techniques to determine endotracheal tube position in a neonatal piglet model. *Pediatric pulmonology*, 48(2), 138-145.
- [7] Lionheart, W. R., Kaipio, J., & McLeod, C. N. (2001). Generalized optimal current patterns and electrical safety in EIT. *Physiological measurement*, 22(1), 85.
- [8] Soulsby, C., Yazaki, E., & Evans, D. F. (2005). Applications of electrical impedance tomography in the gastrointestinal Tract. *Electrical Impedance Tomography: Methods, History and Applications*, 191-207.
- [9] Cheney, M., Isaacson, D., & Newell, J. C. (1999). Electrical impedance tomography. *SIAM review*, 41(1), 85-101.
- [10] Dudykevych, T., Richter, H., Hahn, G., Frerichs, I., Hinz, J., & Hellige, G. (2005). Software and operational concept for EIT-based regional lung function monitoring.
- [11] Suchomel, J., & Sobota, V. (2013). A model of end-expiratory lung impedance dependency on total extracellular body water. In *Journal of Physics: Conference Series* (Vol. 434, No. 1, p. 012011). IOP Publishing.
- [12] Xu, Z., Yao, J., Wang, Z., Liu, Y., Wang, H., Chen, B., & Wu, H. (2018). Development of a portable electrical impedance tomography system for biomedical applications. *IEEE Sensors Journal*, 18(19), 8117-8124.
- [13] Red pitaya user manual, release 0.97. <https://buildmedia.readthedocs.org/media/pdf/redpitaya/latest/redpitaya.pdf>. Last accessed 18 March 2020.
- [14] Sheng, X. Q., & Song, W. (2011). *Essentials of computational electromagnetics* (Vol. 757). John Wiley & Sons.

- [15] Werlberger, M., Pock, T., & Bischof, H. (2010, June). Motion estimation with non-local total variation regularization. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 2464-2471). IEEE.
- [16] Lukashewitsch, M., Maass, P., & Pidcock, M. (2003). Tikhonov regularization for electrical impedance tomography on unbounded domains. *Inverse Problems*, 19, 585-610.
- [17] Brandstatter, B. (2003). Jacobian calculation for electrical impedance tomography based on the reciprocity principle. *IEEE transactions on magnetics*, 39(3), 1309-1312.
- [18] McKean, J. W., & Hettmansperger, T. P. (1978). A robust analysis of the general linear model based on one step R-estimates. *Biometrika*, 65(3), 571-579.
- [19] Turbell, H. (2001). *Cone-beam reconstruction using filtered backprojection* (Doctoral dissertation, Linköping University Electronic Press).
- [20] Behrenbruch, C. P., Petroudi, S., Bond, S., Declerck, J. D., Leong, F. J., & Brady, J. M. (2004). Image filtering techniques for medical image post-processing: an overview. *The British journal of radiology*, 77(suppl\_2), S126-S132.
- [21] TekPower TP3005D-3M DC Power Supply Adjustable 0-30V 0-5A Dual Outputs with USB. <https://tekpower.us/tp3005d-3m.html>. Last accessed 16 March 2020.
- [22] Kaneyama, H. (1991). *U.S. Patent No. 4,985,703*. Washington, DC: U.S. Patent and Trademark Office.
- [23] Formlabs Stereolithography 3D Printers Tech Specs. <https://formlabs.com/3d-printers/form-3/tech-specs/>. Last accessed 17 March 2020.
- [24] OWON PDS5022S Portable Colour Digital Storage Oscilloscope User's Manual. <http://www.saelig.com/downloads/pds5022s.pdf>. Last accessed 16 March 2020.
- [25] VO, V. (1993). Programmable Gain AMPLIFIER.
- [26] Drefs, M. J. (1997). IEC 601-1 electrical safety testing. *Compliance Engineering*, 14(2).
- [27] Shirzadfar, H., Khezri, N., & Khanahmadi, M. (2018). Simulation Based Improvement Technique of Electroencephalography Signals with the LTspice. *SciFed Journal of Bone & Spine*, 1(1).
- [28] Sparkfun. SparkFun Analog/Digital MUX Breakout - CD74HC4067. [https://media.digikey.com/pdf/Data%20Sheets/Sparkfun%20PDFs/BOB-09056\\_Web.pdf](https://media.digikey.com/pdf/Data%20Sheets/Sparkfun%20PDFs/BOB-09056_Web.pdf) Last accessed 19 March 2020.
- [29] Hua, P., Woo, E. J., Webster, J. G., & Tompkins, W. J. (1993). Using compound electrodes in electrical impedance tomography. *IEEE transactions on biomedical engineering*, 40(1), 29-34.
- [30] Community, B. O. (2018). Blender - a 3D modelling and rendering package. Stichting Blender Foundation, Amsterdam. Retrieved from <http://www.blender.org>.

- [31] D O'Neill, R., Chang, S. C., Lowry, J. P., & McNeil, C. J. (2004). Comparisons of platinum, gold, palladium and glassy carbon as electrode materials in the design of biosensors for glutamate. *Biosensors and Bioelectronics*, 19(11), 1521-1528.
- [32] Negi, S., Bhandari, R., Rieth, L., Van Wagenen, R., & Solzbacher, F. (2010). Neural electrode degradation from continuous electrical stimulation: comparison of sputtered and activated iridium oxide. *Journal of neuroscience methods*, 186(1), 8-17.
- [33] Netzer, Y. (1981). The design of low-noise amplifiers. *Proceedings of the IEEE*, 69(6), 728-741.
- [34] Analog Devices. Low Cost Low Power Instrumentation Amplifier AD620. <https://www.analog.com/media/en/technical-documentation/data-sheets/AD620.pdf>. Last accessed 19 March 2020.
- [35] Pallás-Areny, R., & Webster, J. G. (1991). Common mode rejection ratio in differential amplifiers. *IEEE Transactions on Instrumentation and Measurement*, 40(4), 669-676.
- [36] Schmidt, W. C., & Oliverio, J. C. (1969). Electrical noise reduction in industrial control systems. *IEEE Transactions on Industry and General Applications*, (3), 259-265.
- [37] Ott, H. W., & Ott, H. W. (1988). *Noise reduction techniques in electronic systems* (Vol. 442). New York: Wiley.
- [38] Texas Instruments. LM741 Operational Amplifier Datasheet. <http://www.ti.com/lit/ds/symlink/lm741.pdf>. Last accessed 19 March 2020.
- [39] Vauhkonen, Marko & Lionheart, W & Heikkinen, Lasse & Vauhkonen, P & Kaipio, Jari. (2001). A matlab package for the EIDORS project to reconstruct two-dimensional EIT images. *Physiological measurement*. 22. 107-11. 10.1088/0967-3334/22/1/314.
- [40] Hofmann, L., Ebert, M., Tass, P. A., & Hauptmann, C. (2011). Modified pulse shapes for effective neural stimulation. *Frontiers in neuroengineering*, 4, 9.
- [41] Moore, B. C. (Ed.). (1995). *Hearing*. Academic Press.
- [42] Lee Rodgers, J., & Nicewander, W. A. (1988). Thirteen ways to look at the correlation coefficient. *The American Statistician*, 42(1), 59-66.
- [43] Ayati, S. B. (2016). *A portable EIT system for emergency medical care* (Doctoral dissertation, Loughborough University).
- [44] Adler, A., Gaggero, P. O., & Maimaitijiang, Y. (2011). Adjacent stimulation and measurement patterns considered harmful. *Physiological measurement*, 32(7), 731.
- [45] Adler, A., Grychtol, B., & Bayford, R. (2015). Why is EIT so hard, and what are we doing about it?. *Physiological Measurement*, 36(6), 1067.